(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 November 2001 (01.11.2001)

PCT

(10) International Publication Number
WO 01/82080 A2

(51) International Patent Classification[7]: G06F 11/00

(21) International Application Number: PCT/US01/12889

(22) International Filing Date: 20 April 2001 (20.04.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/552,781 20 April 2000 (20.04.2000) US

(71) Applicant: CIPRICO, INC. [US/US]; Suite 60, 2800 Campus Drive, Plymouth, MN 55435 (US).

(72) Inventors: MCMILLAN, Ben, H., Jr.; 125 Marvin Road, Middletown, NJ 07748 (US). DAVIS, Daniel, A.: 33 S. First Avenue, Apartment 2, Highland Park, NJ 08904 (US).

(74) Agent: MACMASTERS, Thomas, L.; Fredrikson & Byron, P.A., 1100 International Center, 900 Second Avenue South, Minneapolis, MN 55402 (US).

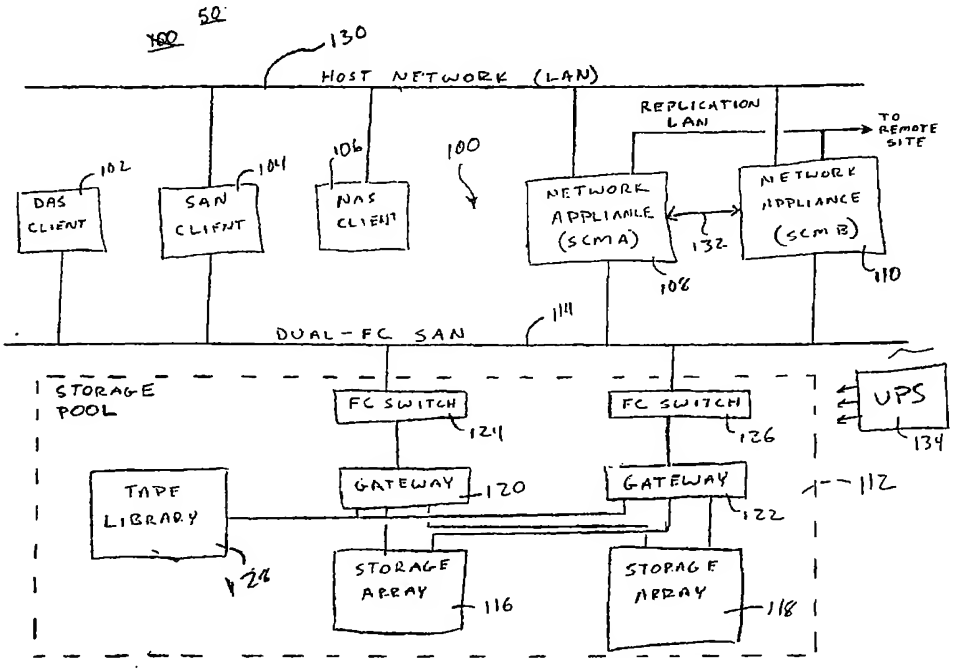
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: FAULT-TOLERANT, HIGH AVAILABILITY NETWORK APPLIANCE



(57) Abstract: A method and apparatus for performing fault-tolerant network computing. The apparatus comprises a pair of network appliances coupled to a network. The appliances interact with one another to detect a failure in one appliance and instantly transition operations from the failed appliance to a functional appliance.

WO 01/82080 A2

FAULT-TOLERANT, HIGH AVAILABILITY NETWORK APPLIANCE

BACKGROUND OF THE DISCLOSURE

1. Field of the Invention

The invention relates to network appliances and, more particularly, the invention relates to a method and apparatus for providing fault-tolerant, high availability network appliances.

2. Description of the Background Art

Data processing and storage systems that are connected to a network to perform task specific operations are known as network appliances. Network appliances may include a general purpose computer that executes particular software to perform a specific network task, such as file server services, domain name services, data storage services, and the like. Because these network appliances have become important to the day-to-day operation of a network, the appliances are generally required to be fault-tolerant. Typically, fault tolerance is accomplished by using redundant appliances, such that, if one appliance becomes disabled, another appliance takes over its duties on the network. However, the process for transferring operations from one appliance to another leads to a loss of network information. For instance, if a pair of redundant data storage units are operating on a network and one unit fails, the second unit needs to immediately perform the duties of the failed unit. However, the delay in transitioning from using one storage unit to another causes some data to not be stored and will be lost.

Therefore, a need exists in the art for fault-tolerant, highly available network appliances that

seamlessly transition from one appliance to another to provide redundant appliance operations.

SUMMARY OF THE INVENTION

The disadvantages associated with the prior art are overcome by the present invention of a method and apparatus for performing fault-tolerant network computing. The apparatus comprises a pair of network appliances coupled to a network. The appliances interact with one another to detect a failure in one appliance and instantly transition operations from the failed appliance to a functional appliance. Each appliance comprises shared configuration and state information such that the transition of services from a failed appliance to an operating appliance is seamless.

In one embodiment of the invention, the apparatus comprises a pair of storage controller modules (SCM) that are coupled to a storage pool, i.e., one or more data storage arrays. The storage controller modules are coupled to a host network (or local area network (LAN)). The network comprises a plurality of client computers that are interconnected by the network.

In operation, the client computers request access to the storage pool via a read or write request that is sent through the network to the SCMs. The storage control managers handle the request by routing the request to an appropriate storage array with the storage pool. If one of the storage controller modules were to fail, the other manager would instantly begin handling requests from the network that would otherwise be handled by the failed appliance. The SCMs share configuration and state information so that the operational SCM can rapidly take over the resources of the failed SCM.

BRIEF DESCRIPTION OF THE DRAWINGS

The teachings of the present invention can be readily understood by considering the following detailed description in conjunction with the accompanying drawings, in which:

FIG. 1 depicts a block diagram of one embodiment of the present invention;

FIG. 2 depicts a block diagram of a pair of storage controller modules;

FIG. 3 depicts a functional block diagram of the pair of storage controller modules;

FIG. 4 depicts a state diagram for a storage controller module;

FIG. 5 depicts a flow diagram of a normal boot process for the pair of storage controller modules;

FIG. 6 depicts a flow diagram of a boot process having a faulted slave storage controller module;

FIG. 7 depicts a flow diagram of a boot process having a faulted master storage controller module;

FIG. 8 depicts a flow diagram of the operation of the pair of storage controller modules when the slave storage controller module fails;

FIG. 9 depicts a flow diagram of the operation of the pair of storage controller modules when the master storage controller module fails;

FIG. 10 depicts a flow diagram of the operation of the pair of storage controller modules when the slave storage controller module resumes operation;

FIG. 11 depicts a flow diagram of the operation of the pair of storage controller module when the master storage controller module performs a failback operation.

FIG. 12 depicts a functional block diagram of the status monitoring system of the pair of storage controller modules; and

FIG. 13 depicts the software architecture for a storage controller module.

To facilitate understanding, identical reference numerals have been used, where possible, to designate identical elements that are common to the figures.

DETAILED DESCRIPTION

One embodiment of the invention is a modular, high-performance, highly scalable, highly available, fault tolerant network appliance that is illustratively embodied in a data storage system. FIG. 1 depicts a data processing system 50 comprising a plurality of client computers 102, 104, and 106, a host network 130, and a storage system 100. The storage system 100 comprises a plurality of network appliances 108 and 110 and a storage pool 112. The plurality of clients comprise one or more of a network attached storage (NAS) client 102, a direct attached storage (DAS) client 104 and a storage area network (SAN) client 106. The plurality of network appliances 108 and 110 comprise a storage controller module A (SCM A) 108 and storage controller module B (SCM B) 110. The storage pool 112 is coupled to the storage controller modules 108, 110 via a fiber channel network 114. One embodiment of the storage pool 112 comprises a pair of storage arrays 116, 118 that are coupled to the fiber channel network 114 via a pair of fiber channel switches 124, 126 and a communications gateway 120, 122. A tape library 128 is also provided for storage backup.

In storage system 100, the DAS client directly accesses the storage pool 112 via the fiber channel network 114, while the SAN client accesses the storage pool 112 via both the LAN 130 and the fiber channel network 114. For example, the SAN client 104 communicates via the LAN with the SCMs 108, 110 to request access to the storage pool

112. The SCMs inform the SAN client 104 where in the storage arrays the requested data is located or where the data from the SAN client is to be stored. The SAN client 104 then directly accesses a storage array using the location information provided by the SCMs. The NAS client 106 only communicates with the storage pool 112 via the SCMs 108, 110. Although a fiber channel network is depicted as one way of connecting the SCMs 108, 110 to the storage pool 112, the connection may be accomplished using any form of data network protocol such as SCSI, HIPPI and the like.

Dependability is one of the major design goals of the storage system 100. Dependability is addressed in four areas, reliability, availability, safety, and security. Through the use of hardware fault tolerance and the deployment of redundant components, the aim of the invention is to eliminate any single points of failure. The availability of the storage system is addressed through a high availability software module (HASM) that provides a high availability environment in which storage applications can operate. This high availability or software fault tolerance aspect is addressed in four sections, fault elimination, fault removal, fault tolerance and fault avoidance techniques. The safety aspects of the system protects the data stored in the storage system so that no data corruption occurs. This is addressed through the use of a redundant array of inexpensive disk (RAID) technique of storing data in the storage arrays. The last aspect of dependability is security that is implemented in the file and directory protection that allows users to protect their data from unauthorized access.

The modularity of the storage system 100 allows a wide range of markets to be addressed by a single product family. This is due to the fact that this storage system can be customized to a customer's needs simply by choosing

the appropriate components. This modularity also allows the customer to grow the storage capabilities as needed. The scalability of the storage system allows a customer to start with a relatively in-expensive single SCM and one storage array configuration, then add additional SCMs and storage arrays to grow the system into an enormous storage system.

The storage system 100 provides substantial flexibility in terms of connectivity configurations. The use of common PCI I/O cards provide connections to direct channel, networks, and SANs. New connectivity options are quickly adopted since only the I/O card and software drivers need be developed and upgraded. The rest of the storage system need not be changed to facilitate implementation of system improvements.

The storage system is a hierarchy of system components that are connected together within the framework established by the system architecture. The major active system level components are:

SCM - Storage Controller Module
SDM - Storage Device Module (Storage Pool)
UPS - Uninterruptible Power Supply
Fibre channel switches, hubs, routers and gateways

The system architecture provides an environment in which each of the storage components that comprise the storage system embodiment of the invention operate and interact to form a cohesive storage system.

The architecture is centered around a pair of SCMs 108 and 110 that provide storage management functions. The SCMs are connected to a host network that allows the network community to access the services offered by the SCMs 108, 110. Each SCM 108, 110 is connected to the same

set of networks. This allows one SCM to provide the services of the other SCM in the event that one of the SCMs becomes faulty. Each SCM 108, 110 has access to the entire storage pool 112. The storage pool is logically divided by assigning a particular storage device (array 116 or 118) to one of the SCMs 108, 110. A storage device 116 or 118 is only assigned to one SCM 108 or 110 at a time. Since both SCMs 108, 110 are connected to the entirety of the storage pool 112, the storage devices 116, 118 assigned to a faulted SCM can be accessed by the remaining SCM to provide its services to the network community on behalf of the faulted SCM. The SCMs communicate with one another via the host networks. Since each SCM 108, 110 is connected to the same set of physical networks as the other, they are able to communicate with each other over these same links. These links allow the SCMs to exchange configuration information with each other and synchronize their operation.

The host network 130 is the medium through which the storage system communicates with the clients 104 and 106. The SCMs 108, 110 provide network services such as NFS and HTTP to the clients 104, 106 that reside on the host network 130. The host network 130 runs network protocols through which the various services are offered. These may include TCP/IP, UDP/IP, ARP, SNMP, NFS, CIFS, HTTP, NDMP, and the like.

From an SCM point of view, its front-end interfaces are network ports running file protocols. The back-end interface of each SCM provides channel ports running raw block access protocols.

The SCMs 108, 110 accept network requests from the various clients and process them according to the command issued. The main function of the SCM is to act as a network-attached storage (NAS) device. It therefore communicates with the clients using file protocols such as

NFSv2, NFSv3, SMB/CIFS, and HTTP. The SCM converts these file protocol requests into logical block requests suitable for use by a direct-attach storage device.

The storage array on the back-end is a direct-attach disk array controller with RAID and caching technologies. The storage array accepts the logical block requests issued to a logical volume set and converts it into a set of member disk requests suitable for a disk drive.

The redundant SCMs will both be connected to the same set of networks. This allows either of the SCMs to respond to the IP address of the other SCM in the event of failure of one of the SCMs. The SCMs support 10BaseT, 100BaseT, and Gigabit Ethernet. The SCMs can communicate with each other through a dedicated inter-SCM network 132 as a primary means of inter-SCM communications. This dedicated connection can employ 100BaseT Ethernet, Gigabit Ethernet or fibre channel. In the event of the failure of this link 132, the host network 130 may be used as a backup network. The SCMs 108, 110 connect to the storage arrays 116, 118 through parallel differential SCSI (not shown) or a fiber channel network 114. Each SCM 108, 110 may be connected through their own private SCSI connection to one of the ports on the storage array.

The storage arrays 116, 118 provide a high availability mechanism for RAID management. Each of the storage arrays provides a logical volume view of the storage to a respective SCM. The SCM does not have to perform any volume management.

The UPS 134 provides a temporary secondary source of AC power source in the event the primary source fails. This allows time for the storage arrays 116, 118 to flush the write-back cache and for the SCMs 108, 110 to perform an orderly shutdown of network services. The UPS is monitored by the SCMS through the serial port or over the host network using SNMP.

FIG. 2 depicts an embodiment of the invention having the SCMs 108, 110 coupled to the storage arrays 116, 118 via SCSI connections 200. Each storage array 116, 118 comprises an array controller 202, 204 coupled to a disk enclosure 206, 208. The array controllers 202, 204 support RAID techniques to facilitate redundant, fault tolerant storage of data. The SCMs 108, 110 are connected to both the host network 130 and to array controllers 202, 204. Note that every host network interface card (NIC) 210 connections on one SCM is duplicated on the other. This allows a SCM to assume the IP address of the other on every network in the event of a SCM failure. One of the NICs 212 in each SCM 108, 110 is dedicated for communications between the two SCMs.

On the target channel side of the SCM, note that each SCM 108, 110 is connected to an array controller 202, 204 through its own host SCSI port 214. All volumes in each of the storage arrays 202, 204 are dual-ported through SCSI ports 216 so that access to any volume is available to both SCMs 108, 110.

Storage Controller Module (SCM) Hardware

The SCM 108, 110 is based on a general purpose computer (PC) such as a ProLiant 1850R manufactured by COMPAQ Computer Corporation. This product is a Pentium PC platform mounted in a 3U 19" rack-mount enclosure. The SCM comprises a plurality of network interface controls 210, 212, a central processing unit (CPU) 218, a memory unit 220, support circuits 222 and SCSI parts 214. Communication amongst the SCM components is supported by a PCI bus 224. The SCM employs, as a support circuit 222, dual hot-pluggable power supplies with separate AC power connections and contains three fans. (One fan resides in each of the two power supplies). The SCM is, for example,

based on the Pentium III architecture running at 600 MHz and beyond. The PC has 4 horizontal mount 32-bit 33 MHz PCI slots. As part of the memory (MEM) unit 220, the PC comes equipped with 128 MB of 100 MHz SDRAM standard and is upgradable to 1 GB. A Symbios 53c8xx series chipset resides on the 1850R motherboard that can be used to access the boot drive.

The SCM boots off the internal hard drive (also part of the memory unit 220). The internal drive is, for example, a SCSI drive and provides at least 1 GB of storage. The internal boot device must be able to hold the SCSI executable image, a mountable file system with all the configuration files, HTML documentation, and the storage administration application. This information may consume anywhere from 20 to 50 MB of disk space.

In a redundant SCM configuration, the SCM's 108, 110 are identically equipped in at least the external interfaces and the connections to external storage. The memory configuration should also be identical. Temporary differences in configuration can be tolerated provided that the SCM with the greater number of external interfaces is not configured to use them. This exception is permitted since it allows the user to upgrade the storage system without having to shut down the system. As mentioned previously, one network port can be designated as the dedicated inter-SCM network. Only SCMs and UPS's are allowed on this network 132.

Storage Device Module (SDM) Hardware

The storage device module (storage pool 112) is an enclosure containing the storage arrays 116 and 118 and provides an environment in which they operate.

One example of a disk array 116, 118 that can be used with the embodiment of the present invention is the

Synchronix 2000 manufactured by ECCS, Inc. of Tinton Falls, New Jersey. The Synchronix 2000 provides disk storage, volume management and RAID capability. These functions may also be provided by the SCM through the use of custom PCI I/O cards.

Depending on the I/O card configuration, multiple Sychronix 2000 units can be employed in this storage system. In one illustrative implementation of the invention, each of the storage arrays 116, 118 uses 4 PCI slots in a 1 host/3 target configuration, 6 SCSI target channels are available allowing six Synchronix 2000 units each with thirty 50GB disk drives. As such, the 180 drives provide 9 TB of total storage. Each storage array 116, 118 can utilize RAID techniques through a RAID processor 226 such that data redundancy and disk drive fault tolerance is achieved.

SCM Software

Each SCM 108,110 executes a high availability software module (HASM), which is a clustering middleware that provides the storage system 100 with a high availability environment. The HASM is a collection of routines and subroutines stored in the memory units of each SCM that, when executed, provides the functionality of the present invention. The HASM allows two SCMs to run in a dual-active symmetrical configuration meaning that either SCM can take over for the other in the event of a SCM fault. The SCM hardware fault tolerance allows a redundant SCM to manage the resources and present the services to the network community of a SCM that has faulted. The software fault tolerance allows all state-less and state-full NAS applications and services such as NFS, HTTP, and CIFS to run in a high availability environment. The software fault tolerance provides a mechanism to detect and recover from

faults in the software whether they evolved from a hardware error or a software design or implementation error. This software allows the SCM to assume the resources and services of a faulted SCM in as transparent a manner as possible within the capabilities of the existing protocols. Clients logged into the faulted SCM must continue to believe they are logged into the same SCM.

The HASM monitors the state of both the local SCM and the remote SCM. The local SCM is continually monitored through the local health monitor (LHM) as well as by the individual applications running on the SCM. If an application finds that a critical error has occurred that prevents the SCM from operating correctly, this application can force the SCM to reboot thus transferring control to the remote SCM. If the LHM has determined that the system is not operating correctly, a surrender to the remote SCM can take place so that the local SCM can reboot and hopefully correct the situation. Attempts to detect internal error conditions must occur as soon as possible. It is the intent of this design to detect and initiate recovery on the order of a few seconds or less. The remote SCM is monitored using status messages that are exchanged between the two SCMs. In one embodiment of the invention, status messages are transmitted across all available network channels. In the event a number of network channels fail, a false detection of a faulted SCM will not occur. The remote SCM is not considered faulted unless it either indicates it has faulted or all status message channels fail. The time-out value is tunable to maximize failover time yet minimize false alarms. If a failover operation is initiated, the procedure must be completed as fast as possible. The primary source of delay in failing over is the checking of the integrity of the file systems (fsck). It is anticipated that this should not exceed a few minutes except in the most strenuous of cases.

If using the host networks as a primary conduit for status message communications is unacceptable, the serial ports may be used in conjunction with the networks to provide a backup for the networks in case the entire host network fails. This would prevent one of the SCMs from thinking the other has failed when in fact this has not occurred.

FIG. 3 is a functional block diagram of the HASM 300 of each of the SCMs. The HASM 300 consists of several major components, each of which work together to create a high availability environment in which embedded NAS applications can run. These components are classified as control, monitor, service, and transition modules. These components include the high availability software controller (HASC) 302, the status monitor (SM) 304, the local health monitor (LHM) 306, the remote SCM communications link (RSCM) 308, the persistent shared object module (PSOM) 310, the shared file manager (SFM) 312, the transition functions 314 (FAILBACK, FAILOVER, FAULT, IMPERSONATE, MASTER, RESET, SHUTDOWN, and SLAVE) and the configuration transaction control module (CTCM) 316. These modules have the following responsibilities:

HASC 302 - High Availability Software Controller - controls the HASM by gathering information from the SM, LHM, and RSCM to determine the state of the SCM if a transition to a new state of operation is required.

SM 304 - Status Monitor - monitors and assesses the health of the remote SCM.

LHM 306 - Local Health Monitor - monitors and assess the health of the local SCM. Has the ability to restart services tasks that have terminated due to nonrecoverable errors.

RSCM 308 - Remote SCM Communications Manager - provides a reliable redundant communications link between the two SCMs for purposes of information exchange and synchronization. This provides the platform over which all inter-SCM communications take place. The RSCM is described in detail in U.S. patent application serial number _____ filed simultaneously herewith (Attorney docket ECCS 007), which is incorporated herein by reference.

PSOM 310 - Persistent Shared Object Manager - provides an object paradigm that allows objects to be distributed between the two SCMs. This module guarantees persistence of objects across power cycles.

SFM 312- Shared File Manager - provides a mechanism for keeping configuration files synchronized between the two SCMs.

CTCM 316 - Configuration Transaction Control Module - provides a configuration transaction paradigm to mark configuration changes to determine configuration state on power up.

FAILBACK module 318 - transitions the SCM from the DEGRADED state to the DAC(master) state.

FAILOVER module 320 - transitions the SCM from the DAC(master) state to the DEGRADED state.

FAULT module 322 - transitions the SCM into the MONITOR mode where configuration corrections and analysis can be made.

IMPERSONATE module 324 - allows a SCM to impersonate the other SCM when the MASTER faults (the SLAVE impersonates the MASTER) and when the MASTER boots and discovers the SLAVE is running in DEGRADED mode (the MASTER impersonates the SLAVE)

MASTER module 326 - transition the SCM from BOOT state to INIT(master) state.

READY module 328 - transitions the SCM from INIT state to DAC state for both the MASTER and SLAVE SCMs.

RESET module 330 - transitions the SCM from the current operational state to the BOOT state.

SHUTDOWN module (included in the RESET module 330)- transition the SCM from the current operational state to the HALTED state.

SLAVE module 332 - transitions the SCM from the BOOT state to the INIT(slave) state.

The HASM 300 allows stateless applications to perform in a high availability manner without modification. Maintenance of any configuration files is handled by the HASM 300 through the system administration module (SAM) (not shown). The SAM will invoke the HASM 300 to keep configuration files synchronized between the two SCMs 108, 110. In fact, the SAM is run as a stateful high availability service on top of the HASM 300. In the case of stateful applications such as the SAM, a set of API calls are provided that allow the state information to be synchronized between the two SCMs. This is provided primarily by the RSCM 308, PSOM 310, SFM 312, and the CTCM 316 as described in detail below with respect to FIG. 13.

The HAS controller (HASC) 302 is the central control module for the HASM. The HASM 300 gathers its information from monitoring and service modules, invokes transition modules to perform transitions from state to state, and communications with the remote HASC to synchronize and invoke operations on the remote SCM. The HASC 302 determines the state of the HASM 300 based on the information it gathers. HASC 302 is responsible for maintaining the current state of the SCM.

The monitoring modules are SM 304 and LHM 306. The SM 304 contains a status generator and status monitor. The status generator generates status messages that are transmitted to the remote SCM. The status monitor is responsible for listening to the status messages and gathering information based on their timely arrivals and or failures to be received at all. One particular technique that can be used to communicate and monitor status information is a heartbeat signal technique that is described in U.S. patent application serial number _____ filed simultaneously herewith (Attorney docket ECCS 006, which is incorporated herein by reference.

The LHM 306 monitors the local SCM looking for discrepancies in its operation. If either of these two modules 304, 306 reports a failure to the HASC 302, the HASC 302 performs the appropriate transition based on what faulted.

The HASC also takes input from the CTCM 316 and RSCM 308. The CTCM 316 is used only during initialization by the HASC 302 to determine the initial state of the HASM 300 software. The RSCM 308 provides error status of the various network channels to determine if the remote SCM is responding to requests properly.

The service modules are the RSCM 308, CTCM 316, PSOM 310, and SFM 312. These modules provide services that are used by components of the HASM 300 and high available

stateful applications. They also provide some feedback on the operation of the remote SCM which is used in conjunction with the SM's information in evaluating the condition of the remote SCM.

The transition modules 318 through 332 provide a capability of transitioning the SCM and storage system from one state to another and are responsible for making storage system transitions from DEGRADED mode to DUAL-ACTIVE CONFIGURATION mode and back to DEGRADED mode as well as for making SCM transitions from MASTER to SLAVE or from SLAVE to MASTER.

In order for the HASM 300 to function correctly, some requirements must be met regarding the functionality of the system in which the HASM 300 is to run:

1. IP aliasing - The network interfaces must be able to respond to multiple IP addresses.
2. Gratuitous ARP notification - This message must be broadcast to notify other computers on the host networks that the MAC addresses has changed for a particular IP address.
3. Initiator ID - The HASM must be able to control the initiator ID of the SCSI HBAs that reside in the SCM. The first SCSI HBA is always designated for exclusive use of the local SCM. Since this is the case, the initiator ID of this SCSI HBA may be set to 7 by default. The remaining SCSI HBAs are used to shared storage on the back-end of the storage system. Due to SCSI requirements, two devices can not have the same SCSI ID on a SCSI bus. Therefore, one of the SCMs must be configured to use an initiator ID of 7 and the other, an initiator ID of 6.
4. All NICs on one SCM must be connected to the NICs on the other SCM.

5. Both SCMs must have access to the same storage pool.
6. The File Systems must be logged structure with checkpointing.
7. File System write caching must be disabled. This prevents cached data from being lost during a SCM failure.
8. All applications running in the HA environment that do not use the HASM services, must be persistent across restarts.
9. Interface and device errors must be reportable to the HASM through a callback function interface or some other similar mechanism.

To achieve seamless transition between SCMs upon the occurrence of a fault, stateful applications require state information to be consistent between SCMs. This allows the remote SCM to have up-to-date information regarding the state of execution of the local SCM. In case, the local SCM faults, the remote SCM has the required information to continue on where the local SCM left off. The requirements of this state information depends upon the specified application. The HASM does provide a set of services that allows state information to be propagated to the other SCM. The stateful applications that require such services are the HTTP server, the NFS server, and the CIFS server. Additionally, the HASM may employ the use of alternate hardware channels to provide low latency state coherency operations.

As mentioned earlier, the system administration module is a stateful application which will use the services of the HASM to maintain state coherency. The SAM uses the services of the RSCM, SFM, CTCM, and PSOM to achieve state coherency. Only the SAM running on the MASTER SCM is allowed to administer changes to the configuration.

Master/Slave Mode Operation of a SCM

A SCM operates in either MASTER mode or SLAVE mode. The SCM running in MASTER mode is the highest authority in the storage system. There can be only one SCM running in MASTER mode. This designation is for internal purposes only and is transparent to the user of the storage system. The operator (system administration) is the only user who is aware of the MASTER and SLAVE mode roles each of the SCMs play. The first SCM configured into the storage system is always designated as the MASTER mode SCM. Any additional SCMs are designated as SLAVE mode SCMs. There may be one or more SLAVE mode SCMs.

Upon power up, a properly running redundant SCM configuration runs in their configured operational mode (MASTER or SLAVE). The SCM configured for MASTER mode runs in MASTER mode and the SCM configured for SLAVE mode runs in SLAVE mode. If the MASTER mode SCM fails, the SLAVE mode SCM must transition itself into the MASTER mode state. The configured SLAVE mode SCM will continue to run in MASTER mode until it is powered down or faults. The configured MASTER mode SCM upon reboot, will configured itself to run in SLAVE mode until it is powered down or the remote SCM faults.

The applications that execute in the HASM will assume different roles depending on if the SCM on which it resides in executing in MASTER or SLAVE mode. Not all applications behave in this manner. For example, the status monitor runs identically regardless of which mode the SCM is running in. The name_OpMode() function is used to notify the software module as to which mode the SCM is operating in and when a transition from one mode to another occurs.

The storage administration module (SAM) only allows administration on the MASTER mode SCM. All administration commands from the network community must be directed to the

MASTER mode SCM. A request to the SLAVE SCM will return a redirection request to the MASTER SCM.

High Availability Software Controller (HASC)

The HASC is responsible for determining the state of operation of the SCM and performing state transitions based on the information gathered by the SM, the LHM, and the CTCM. The SM and RSCM provide the HASC with status information about the remote SCM, the LHM provides information regarding the health of the local SCM, and the CTCM returns information regarding the state of the configuration information on initial powerup. The HASC gathers information from these sources to determine the correct operational state of the SCM. If the HASC determines that a state transition is required, it will invoke the appropriate routines to correctly perform the transition. For example, if the SCM is running in DAC mode and the SM reports back that the remote SCM is no longer generating or responding to status messages, the HASC will call the FM to perform a failover procedure. The HASC will transition the SCM from DAC mode to degraded mode operation. The HASC then invokes the name_OpMode() functions for all affected software modules to transition the operation of the SCM from DAC mode to degraded mode.

State and Transitions

An SCM runs in context of one of several different states. At the highest operating level, the SCM is either powered on or powered off. If powered on, the SCM is in one of three major state, initializing state, steady state, or fail state. The valid initialization states are BOOT and INIT. The valid fail states are HALTED, and MONITOR. The valid steady states are DAC and DEGRADED.

The SCM may also transition from state to state using the following transitions: POWER ON, READY, FAILOVER, FAILBACK, IMPERSONATE, SHUTDOWN, REBOOT, and POWER OFF.

On the order in which the various software modules are invoked to transition the SCM from state to state, it is recommended that all HASM specified modules be called prior to the software applications running in the HASM environment. This will facilitate the modularization of the HASM software such that it can be more easily ported to other hardware platforms in the future.

These opmodes, states and transitions are explained below:

SCM OpModes:

MASTER - the SCM is operating in the role as MASTER. It is the highest authority SCM in the storage system. This indicates that this SCM is the point of administration and control for the storage system. Only the MASTER controller can change the configuration. The EMM monitors the storage pool, the event log controls the storage of events, the PSOM controls the locking of persistent shared objects.

SLAVE - the SCM is operating in the role as SLAVE. This means that this SCM must follow the commands of the MASTER SCM.

SCM States:

POWERED OFF - the SCM is not powered on. It performs no operation in this state. It transitions to this state when powered is removed from the SCM. Its transitions out of this state to the INIT state after the power is applied to the SCM.

BOOT - [initialization state] - the SCM runs diagnostics and boots up the SCM off the internal hard drive of the SCM. While the system is operating in this state, name_Init() and name_Test() calls are made to initialization and test the various software modules that comprise the set of storage applications that interacts with the HASM. The SCM then determines its operational mode and transitions to the INIT state through either the MASTER or SLAVE transition.

INIT(master/slave) - [initialization state] - the SCM enters this mode for initialization purposes. This state allows the SCM to discover the remote SCM, establish contact with it to determine what its next course of action will be and join with it to create a highly available storage system. Upon completion of this step, the SCM will enter steady state in DAC mode. If the remote SCM is in DEGRADED mode and the local SCM is configured to run in MASTER mode, the local SCM must first transition to SLAVE mode and rerun its initialization process. If the remote SCM is in DEGRADED mode and local SCM is running in SLAVE mode (whether configured or current), the local SCM will remain in this step until the remote SCM has transitioned to DAC mode. The local SCM will then complete its initialization by initializing its resources and offering its assigned services to the network community.

DAC(master/slave) (Dual-Active Configuration) - [steady state] - this is the normal mode of operation of the SCM. This indicates that both SCMs are managing there assigned resources and are offering the services for which they were each configured. An SCM in this

state continually monitors the remote SCM. An SCM will run differently in this state depending on whether it is running in the MASTER role or the SLAVE role. When entering steady state for the first time, a SCM always assumes it is running in DAC mode. Anytime after this event, if the SLAVE mode SCM (currently running in DAC(slave) state) realizes that the remote SCM is not present, the SCM transitions to MASTER mode (now running in DAC(master) mode) and then performs a FAILOVER transition is performed to enter DEGRADED mode. If the MASTER mode SCM detects that the remote SCM is not operating, its can directly perform a FAILOVER transition to DEGRADED mode.

DEGRADED - [steady state] this indicates that the remote SCM is not operational requiring the local SCM to perform the duties of the faulted SCM. The local SCM is always running in MASTER mode while executing in a state of DEGRADED. The local SCM will remain in this state until the remote SCM begins responding to the local SCM's heartbeat messages. The local SCM will perform a transition to DAC mode only after determining that the remote SCM is operational. While in this state, the SCM continually attempts to contact a remote SCM

MONITOR - [fail state] - the SCM enters this mode if the configuration information is corrupted or does not make sense. It provides an opportunity for the operator to correct the situation. While in this state, the SCM is available for administration through the CML only. This state is entered from the INIT state. This transition, FAULT, shuts down all HASM operations, the transition,

name_OpMode(SCM_OPMODE_MASTER, SYSCONFIG_DEGRADED) is invoked.

FAILBACK - The SCM has detected that the remote SCM is now operational when this transition is performed. This transition will perform the necessary tasks to allow the local SCM to transfer control of the remote-owned resources back to the remote and to allow the remote-offered services to be re-continued by the remote SCM. On this transition, name_OpMode (SCM_OPMODE_MASTER, SYSCONFIG_DAC) is invoked.

IMPERSONATE - The SLAVE SCM impersonates the MASTER SCM if the MASTER SCM has faulted or the MASTER SCM impersonates the SLAVE SCM if the configured SLAVE mode SCM is running in DEGRADED mode. On SLAVE to MASTER transitions, name_OpMode(SCM_OPMODE_MASTER, SYSCONFIG_DAC) is invoked, On MASTER to SLAVE transitions, name_OpMode(SCM_OPMODE_SLAVE, SYSCONFIG_ASSUMEDAC) is invoked.

SHUTDOWN - The SCM transitions from an operational state to a halted state. The SCM will remain in the HALTED state until the SCM has been powered off and back on or if the reset button on the SCM is pressed. This transition can be executed on command through the SAM or through receiving notice that AC power has been lost and we are running on our alternate power source (UPC). On this transition, name_Stop() followed by name_ShutDown() is invoked.

RESET - The SCM transitions from an operational state to the BOOT state.

POWER OFF - The power switch is turned off or AC power is lost to the SCM.

FAULT - A transition to MONITOR mode is required. This transition invokes, name_Stop() invocations to halt operation of the SCM. Only the administration interface and any modules required of it will be left in an operating condition such that the SCM can be administered. The PSO will need to break from the cluster and remain in single SCM mode.

FIG. 4 depicts a state diagram 400 for the storage system 100. The SCM starts in a powered off state 402, OFF. It transitions to a powered on state 404 through a POWER ON transition 406. The SCM may enter the powered off state 402 at anytime through a POWER OFF transition 408. Once the SCM enters the powered on state 404, it jumps to the initialization state 410. Upon entering the initialization state 410, the SCM jumps to the BOOT state 412 and begins booting up the SCM. Upon completion of the BOOT, the SCM will transition to the INIT state 414 dependent upon which configured operational mode the SCM is configured for. Either the SCM transitions to the INIT(master) state 414 or the INIT(slave) state 416 depending on if it is configured to run in the MASTER mode or the SLAVE mode, respectively.

Once initialized, the master mode SCM transitions to a DAC master state 418 where it will operate until one of the SCMs fail. The slave mode SCM transitions from the initialization state 416 to the DAC slave state 420 where it will operate until one of the SCMs fail. If the master mode SCM fails, the slave mode SCM transitions along the IMPERSONATE transition 422 to enter a DAC master state 418. Then, because the master SCM must provide services to all the clients (i.e., the master processes requests for both

SCMs), the master SCM transitions to the DEGRADED state 424. If the slave mode SCM fails, the master mode SCM transitions to the DEGRADED state 424. When a failed SCM recovers, the master SCM transitions from the DEGRADED state 424 to the DAC master state. A recovered SCM always boots to the slave state 420.

The fail state 426 contains sub-states monitor 428 and halted 430. From steady state operation, a shutdown transition 432 causes the system to enter the halted state 430. From the fail state 426, if user chooses to restart the SCM or if the reset button is pressed, the SCM transitions along a reboot path 434 to the initialization state 404. If, in the initialization state 404, a fault occurs, the SCM transitions along path 436 to the fault state 426. In the monitor state 428, a minimal amount of functionality is enabled to allow system diagnostics to be performed on a failed SCM, i.e., read the event log, access certain files, and the like.

Normal Boot

FIG. 5 depicts a flow diagram 500 of the boot sequence for a normal system boot.

Initially, at steps 502 and 504 both SCMs are off and are powered on at the same time at steps 506 and 508. Both SCMs boot at steps 510 and 512. After booting, at steps 514 and 516, the SCMs examine their respective configuration files to see which operational mode they were configured to run in. The SCMs then, at steps 518 and 520, run in the appropriate INIT state. During initialization, NOP and GETSTATE messages are exchanged between the two SCMs in order to determine the existence, operability and operational mode of the remote SCM. At steps 522 and 524, READY transition is then performed which brings the SCMs into DAC mode. The storage system is now ready to accept

and complete requests from the network community, i.e., SCM A is now operating in master mode 526 and SCM B is operating in slave mode 528.

Booting with Faulted SLAVE SCM

FIG. 6 depicts a flow diagram 600 of the boot sequence with a SLAVE SCM that is faulted.

The operation SCM (SCM A) boot is normal until it reaches the INIT(master) state. The operational SCM discovers that the remote SCM (SCM B) is not responding. SCM A then decides after a period of time to continue with its boot process. Once the boot process reaches DAC(master) mode 526, the SM detects that no status messages are received or transmitted so it initiates a FAILOVER operation at step 602. The operational SCM transitions at step 604 to a DEGRADED state where it provides the services for both SCMs to the network community.

Booting with a faulted MASTER SCM

FIG. 7 depicts a flow diagram 700 of the boot sequence with a MASTER SCM (SCM A) that has faulted.

This sequence is similar to the previous scenario except that the configured MASTER SCM is faulted. The configured SLAVE SCM must perform at step 702 an IMPERSONATE transition to change from DAC(slave) to DAC(master) mode (step 704) before proceeding with the FAILOVER procedure at step 706. The operational SCM (SCM B) then operates at step 708 in the degraded state.

SLAVE SCM Faults

FIG. 8 depicts a flow diagram 800 of a process that occurs when the SLAVE SCM (SCM B) faults forcing the MASTER SCM (SCM A) to perform a failover operation.

If the SLAVE SCM faults (e.g., status messages cease to be generated) at step 806, the MASTER SCM simply performs a FAILOVER transition at step 802 to change the state of the MASTER SCM from DAC(master) to DEGRADED. At step 804, the MASTER SCM now provides services for both SCMs to the network community. Since the operational SCM has shared the failed SCMs configuration files and application state information, the failover transition is seamless.

MASTER SCM faults

FIG. 9 depicts a flow diagram 900 of a process that occurs when the MASTER SCM (SCM A) faults forcing the SLAVE SCM (SCM B) to impersonate the MASTER SCM and perform a failover operation.

This scenario is similar to the previous one except that the SLAVE SCM must transition to an operational mode of MASTER before performing the FAILOVER procedure. As such, when no status messages are detected because the master SCM has faulted at step 902, the slave SCM transitions at step 904 to the impersonate state. Then, at step 906, the operational SCM transitions into DAC master state. Finally, at step 908, the SCM transitions through a failover transition to a degraded state 910.

SLAVE SCM Resumes

FIG. 10 depicts a flow diagram 1000 of a process that occurs when the MASTER SCM (SCM A) is running in DEGRADED mode while the SLAVE SCM (SCM B) resumes, the MASTER SCM transitions to DAC(master) mode while the SLAVE SCM

transitions to DAC(slave) mode thus bringing the system back to DAC mode.

At step 804, the MASTER SCM (SCM A) is running in DEGRADED mode, while the SLAVE SCM (SCM B) is booting and initializing as described with respect to FIG. 5 above. When the slave SCM sends a ready signal to the master SCM during the INIT state, the master transitions at step 1002 from the degraded state 804 to the DAC master state 526. Once in the DAC master state and the resources that are used by the slave SCM have been released by the master SCM, the master SCM instructs the slave SCM to proceed. The slave SCM then transitions at step 524 to the DAC slave state 528.

MASTER SCM Resumes

FIG. 11 depicts a flow diagram 1100 of a process that occurs when the configured SLAVE SCM is running in DEGRADED mode while the configured MASTER SCM resumes operation. At step 1110, the operational SCM is operating in a degraded state. The failed SCM boots as described in FIG. 5 using steps 502, 506, 510, 514 and 518. Note that SCM A is configured to be a master SCM, so when booting, this SCM initializes as a master SCM. During the INIT state, SCM A requests the state of SCM B and is informed of its degraded state. SCM then transitions at step 1102 to impersonate a slave SCM and then at step 1104 enters the INIT(slave) state. During the INIT(slave) state, SCM A informs SCM B that SCM A is ready to operate. SCM B, at step 1112, begins a failback transition to provide SCM A with information it needs to carry on with operations previously handled by SCM B. Additionally, SCM B relinquishes control of certain resources for SCM A to use. SCM B then enters the DAC(master) state 1114. SCM B sends a proceed command

to SCM A and SCM A transitions, at step 1106, to the DAC(slave) state 1108.

Initialization

Upon initialization, the HASC needs to establish the operational mode of the SCM (MASTER or SLAVE), the operational mode of the storage system (DAC or DEGRADED), and the integrity of the configuration information. The operational mode of the SCM is first assumed to be the configured operational mode. This assumption is made until otherwise changed, i.e., the SCM will run in this assumed operational mode until the SCM establishes communications with the remote SCM and the operational state of the remote has been established. If the remote SCM is in degraded mode, the local SCM must run in SLAVE mode regardless of its configured operational mode. The HASC will invoke the IMPERSONATE transition function to transition the SCM into SLAVE mode if it is configured to operate in MASTER mode. If the remote SCM is booting, the local SCM uses its configured operational mode unless it conflicts with the remotes. In this case, a configuration error has occurred on the part of the system administrator. The SCMs will enter MONITOR mode to allow the system administrator to correct the problem.

The operational mode of the storage system is determined solely on the condition of the remote SCM. If the remote SCM is alive and operating correctly, a mode of DAC is established. If the remote SCM cannot be contacted or it is not responding correctly, a mode of DEGRADED is established.

The configuration information is established by examining the information from the CTCM regarding the state of the SCM configuration. The CTCM indicates if the

configuration information is properly synchronized and if it is consistent with the information stored on the logical storage devices. It will determine if the current SCM contains the latest configuration information and if not, will allow the SCM to retrieve the latest configuration information from the replicated configuration database after which a reboot operation will occur. The replicated configuration database is disclosed in detail in U.S. patent application serial number _____ filed simultaneously herewith (Attorney docket ECCS 008), which is incorporated herein by reference.

The HASC will invoke name_OpMode() function in the appropriate software modules after the operational mode of the SCM and the operational mode of the storage system has been established.

Committing Suicide

If the LHM has determined that the local SCM is no longer capable of operating correctly in its current state, the LHM will request that the HASC commit one of several types of suicide. If the LHM was determined that it is a software condition caused by a design or implementation fault, the HASC will simply reboot the SCM. The remote SCM will takeover control of the local SCM's resources and services. If the LHM has determined that a more permanent condition has arisen such as a hardware failure, the local SCM will be disabled until a power cycle is applied and the remote SCM will takeover the resources and services of the local SCM as well.

Status Monitor

The SM is responsible for monitoring the status messages of the remote SCM to determine if the remote SCM

is alive and operating properly. If the SM determines that the remote SCM is not operating correctly, it will notify the HASC to initiate a failover operation. The SM employs redundant channels in order to transmit and receive status messages.

FIG. 12 depicts a block diagram of an illustrative embodiment of a status monitor 1200. Specifically, he SM is divided into a client 1202 and server 1204 task. Each SCM employs both a client and a server. The client 1202 comprises a status message generator 1206, a TCP/IP stack 1208, a plurality of NIC drivers 1210 and a plurality of NICs 1212. The status message 1202 client is responsible for issuing status messages on a periodic basis. The messages are coupled through a plurality of sockets 1214 to be broadcast on a plurality of network paths 1216. This client task status issues these messages once every second across all available network channels to the server 1204 in the remote SCM. This allows a verification of all network channels to ensure that both SCMs are connected to all networks. This is important because, if a SCM failure occurs, the remaining SCM must have access to all resources connected to the failed SCM. The client 1202 also updates the status information which contains the status of all the network channels.

The server 1204 comprises a status message receiver 1218, a status API 1220, a status analyzer 1222, a fault analyzer 1224, a status information database 1226, and a network communications portion 1228. The network communications portion 1228 comprises a plurality of sockets 1230, a TCP/IP stack 1232, a plurality of NIC drivers 1234 and NICs 1234. The server 1204 is an iterative server and listens for status messages on the set of sockets 1230 to all the available network interfaces and performs analysis on the state of the various network channels over which status messages are received. The

server 1204 updates the status information database 1226 every time that a status message is received from the client 1202 running on the remote SCM. The status information database 1226 contains the current state of each network port. The status analyzer 1222 checks the status information database 1226 on a periodical basis. The status analyzer 1222 is looking for network ports that are not being updated. An un-updated network channel status indicates that some sort of fault has occurred. The status analyzer 1222 calls the fault analyzer 1224 to analyze the situation. The fault analyzer 1224 is also responsible for updating the network port objects through a socket 1238 coupled to the TCP/IP stack 1232 and the remote SCM configuration object. The status API 1220 allows the status of the status monitor 1220 to be returned. Information regarding the status monitor 1200 as well as the network channel state and remote SCM state are available.

If no status messages are being received from the remote SCM, the SCM assumes that the remote SCM has failed. The HASC is notified of this condition.

If one of the host network ports is not working properly, status messages issued over the inoperative channel are not received by the status server message. An event is logged to an event notification service. If the dedicated SCM channel is not operational, no actions are taken other than the notification of the event. If one of the Host network connections has become inoperative, the status monitor 1200 in conjunction with the remote SCM's status monitor attempt to determine the location of the fault as the local SCM's network port, the cabling between the local SCM and the network, the network is down (hub has failed), the remote SCM's network port has failed, or the remote SCM's network cable has failed.

The status monitor communicates through a special RSCM interface designed for the status monitor. This special interface allows better control over the communications channel so that the status monitor can better perform its job function.

The server will need to wait on several sockets using the SELECT command. Every time a status message is received, the sequence number is stored and the count information is incremented by the difference between the current sequence number and the last sequence number. The time-out value is 1 second. Every second, the status analyzer function is run to adjust the status information. It decrements the network channel count information by 1. If the count information hits 0, this indicates the network channel is not working. The starting value for the network channel count information will start at 10. It may need to be adjusted later.

The API allows another task to inquire about the status of the network connections and the remote SCM. The API returns a GOOD/BAD indication of each network connection as well as for the remote SCM. Statistically information must also be returned regarding number of packets send/received, number of missing packets and on which network connections.

One embodiment of a status monitor is described in U.S. patent application serial number _____ filed simultaneously herewith, (Attorney docket ECCS 006), which is incorporated herein by reference. The present invention may utilize the foregoing status monitor technique or any other technique that facilitates identification of a faulted remote SCM.

Local Health Monitor

The local health monitor is part of the HASM that monitors and assesses the health of the local SCM. It basically looks at the current state of operation and ascertains the health of the SCM. If the LHM determines that the SCM is not running in a sane state, its logs the condition to the event notification service and then notifies the HASC that it recommends that the SCM surrender its resources and services to the remote SCM and to initiate a reboot operation (suicide module). The reboot operation is intended to reinitialize the SCM in hopes that the problem could be corrected through this re-initialization process. If an application task is running and it recognizes an unrecoverable error condition, it has the ability to call the LHM. Additionally, if a fault condition occurs such as a divide by zero error, the LHM will intercept the fault and forward the request to the HASC which will reboot the system.

The LHM also gathers environment status information from the EMM and uses this in its determination of the health of the local SCM. If the EMM has determined that the system is in danger of overheating, it can initiate a shutdown operation to prevent data from being corrupted.

The LHM is also able to monitor all the services tasks that are running and have the ability to restart them in case the server task terminates due to an unrecoverable error.

Applications that upon detection of an unrecoverable situation that requires re-initializing the SCM, the PANIC function may be called.

On a periodic basis, the LHM generates a wellness message which is transmitted through the event notification service. This wellness message is intended to convey that

the storage system is working normally. The system administration module supplies an API that administers the interval upon which this message is transmitted.

SCM Software Architecture

FIG. 13 diagrammatically depicts the relationship of the main system software components. The RSCM 308 sits on top of the TCP/IP stack 1302 which it uses as a transport for communicating with a remote SCM. The RSCM 308 consists of two layers, the remote SCM communications API 1308 and the redundant link management 1310.

The Remote SCM Communications Manager (RSCM) is responsible for maintaining a reliable communications link with the remote SCM. This module provides a service which allows other software modules and applications in the software architecture to communicate with the remote SCM. This module also provides a redundant link management layer which is responsible for managing the status information regarding the various channels used for communications between the SCMs. The RSCM provides a variety of communications mechanisms. It provides synchronization primitives and synchronous and asynchronous message passing. The RSCM module provides the capability of using physical interfaces such as serial (RS232-C), disk block communications, fibre channel, memory mapped (VIA), and more.

The API 1308 provides an abstraction layer to the addressing problems in working with redundant communications links between a client and a server. This API 1308 simplifies the process of establishing a communications channel, reading and writing data, and terminating a connection to a peer application running on the remote SCM. All common network programming APIs (client/server applications 1312 and status monitor

applications 1314) are encapsulated within this layer. This allows errors to be recorded internally and decisions to be made regarding the choice of network channels without intervention or knowledge of the invoking task. This status information is passed on down to the redundant link manager 1310 that records and collates this information.

The redundant link management (RLM) is responsible for determining the configuration of networks between the SCMs and providing a decision function that decides which network port to use when a channel is opened to a remote server. The RLM updates information related to the configuration of the network ports, their status, and error statistics. This information is stored in the RLM module.

The RLM is responsible for recording error statistics regarding the various network channels and providing a selection criteria for selecting a communications channel for a network application. The RLM is coupled via a socket 1316 to a file system layer 1305 (stack OS) (the TCP/IP stack 1302 and the NIC drivers 1304). Underlying the file system layer 1305 is an operating system (VxWorks) 1306. Anytime an error occurs on a socket connection to the remote SCM, the RLM looks up the sockets IP address and use it to determine which network channel is having the problem. The RLM then updates the NIC statistics for the appropriate NIC. Anytime that a client application needs to open a socket to the remote SCM, this module will determine which network channel is the best one to use. The statistics for each network port are stored in the appropriate NIC object which is managed by the RLM module. These statistics are updated by both the RSCM in case of an error and by the SM. The RSCM records only operational errors whereas the SM records both operational and time domain errors (status message not received in time).

Persistent Shared Objects (PSO) are used as a paradigm for sharing data between the SCMs. A PSO is an object

whose value is persistent across power cycles and that is accessible from any SCM in the storage system cluster. The PSO manager 310 is responsible for maintaining the coherency of the information between the various SCMs. Persistence is implemented by storing the PSO in the root file system of each SCM. The object name is identical regardless of which SCM accesses it.

The PSO manager 310 will sit on top of the remote SCM communications module which it uses for all communications between itself and the remote SCM.

A shared object is referenced by its name or its ObjectID. The name is an ASCII string that contains the name of the object in english. The ObjectID is an opaque value assigned by the PSO Manager and is used to reference the shared object for all operations. The ObjectID is actually a pointer to the object. The ObjectID allows the object to be quickly referenced. The ObjectID is not guaranteed to be identical from power cycle to power cycle. It is also not guaranteed to be the same from SCM to SCM.

A shared object must be created before it can be used. The creation process allocates the resources required by the object and return an ObjectID to the creation task. If a shared object is already created, the calling task should call the pso_Exist() function first to see if the object exists already. A shared object has an attribute called persistence. If a shared object is persistent, this indicates that the shared object maintains its existence through storage system power cycles. If a shared object is not persistent, it must be re-created every time the storage system is powered up.

Before a shared object is to be written, it must be locked for exclusive access. This lock extends through the set of SCMs participating in the shared object global name space. The lock guarantees that the lock owner has the exclusive right to modify the shared object. Upon

completion of the modifications to the shared object, the task must unlock the object to allow other local or remote tasks to gain the right to modify it.

A copy of the shared object will exist on all SCMs. Anytime that value of a shared object is changed, the changed object must be distributed to all the other SCMs participating in this distributed application. The PSO Manager maintains a database of the currently operating SCMs. This database is used as the distribution list when a shared object needs to be updated. All SCMs must be aware of all other participating SCMs.

A shared object must possess a locking mechanism that prevents multiple writers from updating the object incorrectly. This locking mechanism will guarantee mutual exclusion and must be able to lock across SCM extents. A binary semaphore must be associated with each copy of a shared object.

The shared file module 312 is responsible for synchronizing the information stored in two different files on one or two systems. Whenever a file is written by the MASTER SCM, it is updated by calling the appropriate APIs in the SFM 312.

The configuration transaction control module (CTCM) 316 is responsible for maintaining the integrity of the configuration of the storage system. Its main responsibility is to maintain the shared replicated configuration database stored on the two SCMs and the storage arrays. The CTCM 316 is not responsible for maintaining the actual configuration files, but acts as a transaction wrapper around which configuration changes take place. Only the SCM running in MASTER mode is allowed to perform a configuration transaction. The SLAVE mode SCM must run under the direct supervision of the MASTER SCM. Only the MASTER can dictate to the SLAVE as to when its own timestamp files can be updates.

The CTCM 316 is invoked anytime one of the SCM configuration files is updated. Most of these calls originate from the SAM.

The configuration information is stored both on the SCM's internal hard drive (referred to as the local configuration database) in its usable format as well as on the private extent of a selected group of logical storage devices (referred to as the shared replicated configuration database). The shared replicated configuration database is considered the primary source of configuration information. It is the responsibility of the CTCM 316 to maintain consistency of information in the local and shared replicated configuration database such that the latest version of the configuration information can always be reliably extracted.

The CTCM 316 is able to ascertain on powerup, if the local configuration information is correct or not. If it is not the latest version or if the configuration information is corrupt, the CTCM can retrieve a copy of the latest configuration from the shared replicated configuration database and correct the corruption. This is implemented by performing a restore operation of a valid archive found on the storage array.

Although various embodiments which incorporate the teachings of the present invention have been shown and described in detail herein, those skilled in the art can readily devise many other varied embodiments that still incorporate these teachings.

What is claimed is:

1. A network appliance for providing network services to a plurality of clients comprising:

first means for communicating with said clients;

second means for communicating with network service equipment;

means, coupled to said first and second communicating means, for storing state and configuration information regarding a remote network appliance; and

means, coupled to said storing means, for utilizing said state and configuration information regarding said remote network appliance to cause said apparatus to perform services that are provided by said remote network appliance.

2. The network appliance of claim 1 wherein said first means for communicating comprises a network interface card and a network interface card driver.

3. The network appliance of claim 1 wherein said second means for communicating comprises SCSI channel equipment.

4. The network appliance of claim 1 wherein said second means for communicating comprises fiber channel equipment.

5. The network appliance of claim 1 further comprising a storage pool for storing data that can be accessed by the clients.

6. The network appliance of claim 1 further comprising means for monitoring a status of said remote network appliance.

7. The network appliance of claim 6 further comprising means for impersonating said remote appliance when said status indicates said remote network appliance is not operating properly.

8. The network appliance of claim 1 further comprising a means for monitoring a status of said network appliance and if said network appliance is not operating properly, re-booting said network appliance.

9. A storage system comprising:

a first storage controller module couple to a network;

a second storage controller module coupled to said network;

a storage pool coupled to said first and second storage controller modules;

where said first storage controller module stores configuration and state information about said second storage controller module and said second storage controller module stores configuration and status information about said first storage controller module.

10. The storage system of claim 9 wherein said first storage controller module comprises a status monitor that monitors the status of the second storage controller module, and said second storage controller module comprises a status monitor that monitors the status of the first storage controller module.

11. The storage system of claim 9 wherein said storage pool comprises a first storage array coupled to said first and second storage controller modules, and a second storage array coupled to said first and second storage controller modules.

12. The storage system of claim 11 wherein said storage arrays are coupled to said storage controller modules using either SCSI connections or a fiber channel network.

13. A method of operating a storage system having a first and second storage controller modules coupled to a storage pool, said method comprising:

executing an initialization state to boot a first storage controller module into a master state; and

executing an initialization state to boot a second storage controller module into a slave state.

14. The method of claim 13 further comprising:

operating said first and second storage controllers in a steady state until a fault is detected in one of said first or second storage controller modules;

causing an operational storage controller module to enter a degraded mode, wherein said operational storage controller module performs functions that are otherwise performed by said faulted storage controller module;

rebooting said faulted storage controller module.

15. The method of claim 14 further comprising:

causing said faulted storage controller module to reboot in a slave mode;

releasing resources from said operational storage controller module to enable said rebooted storage controller module to control said resources.

16. The method of claim 15 wherein said first and second storage controller modules share status and configuration information.

17. The method of claim 15 wherein said first and second storage controller modules analyze status information from a remote storage controller module.

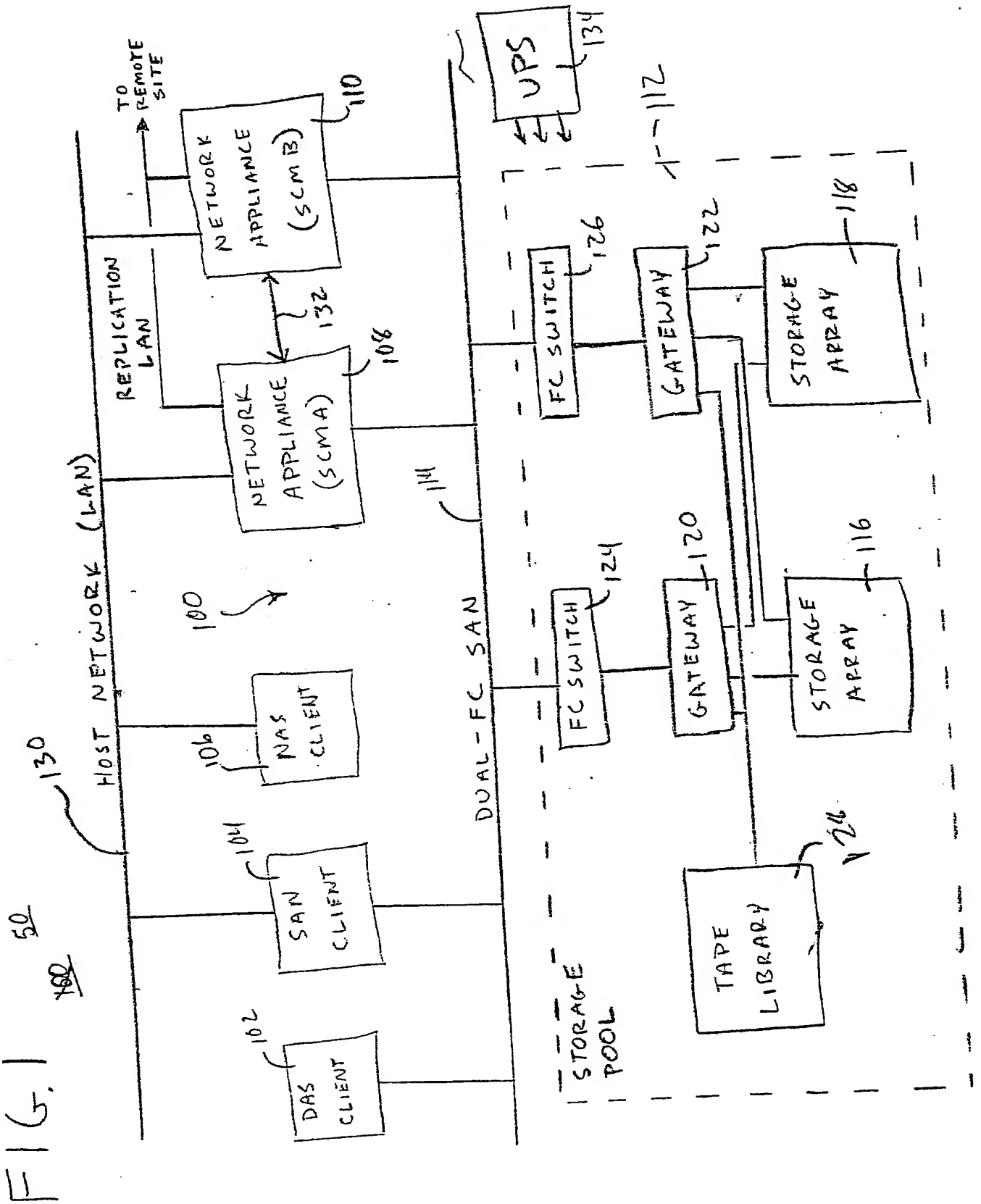

TO REMOTE SITE
NETWORK APPLIANCE (SCM B)
110
REPLICATION LAN
132
NETWORK APPLIANCE (SCM A)
108
UPS
134
112
HOST NETWORK (LAN)
130
100
141
DUAL-FC SAN
FC SWITCH
126
GATEWAY
122
STORAGE ARRAY
118
106
NAS CLIENT
104
SAN CLIENT
FC SWITCH
124
GATEWAY
120
STORAGE ARRAY
116
102
DAS CLIENT
STORAGE POOL
TAPE LIBRARY
114

FIG. 1

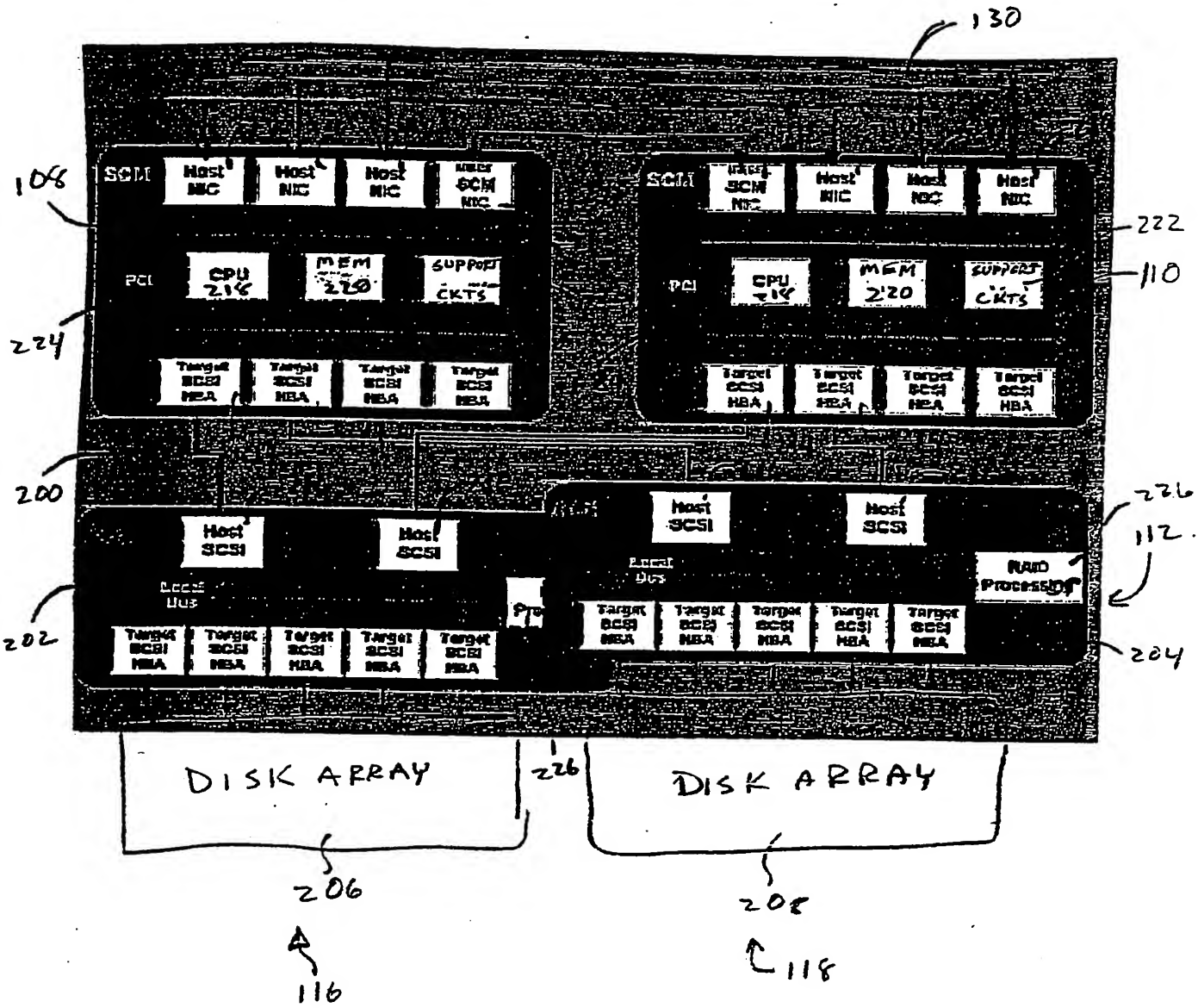


FIG 2

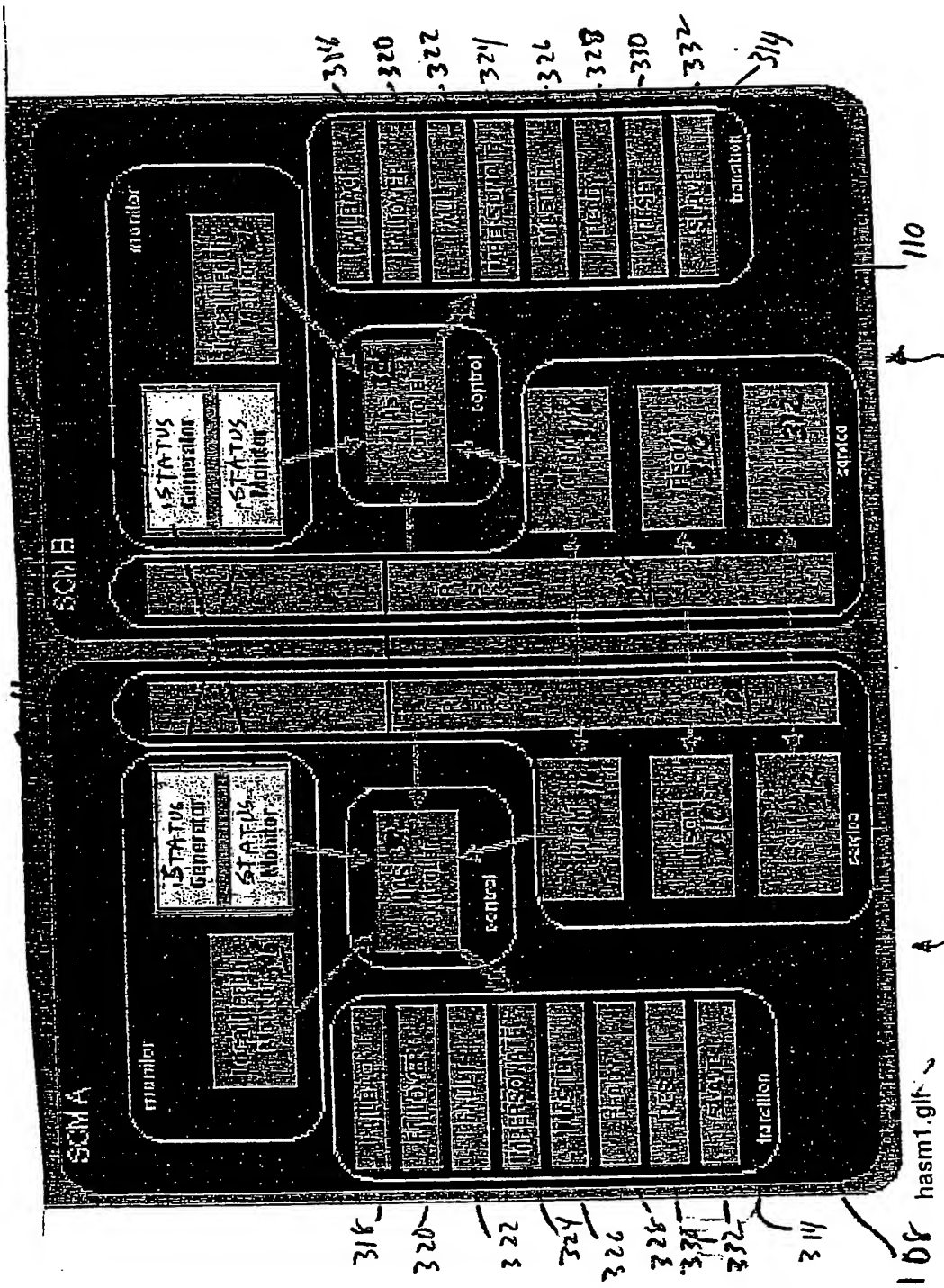
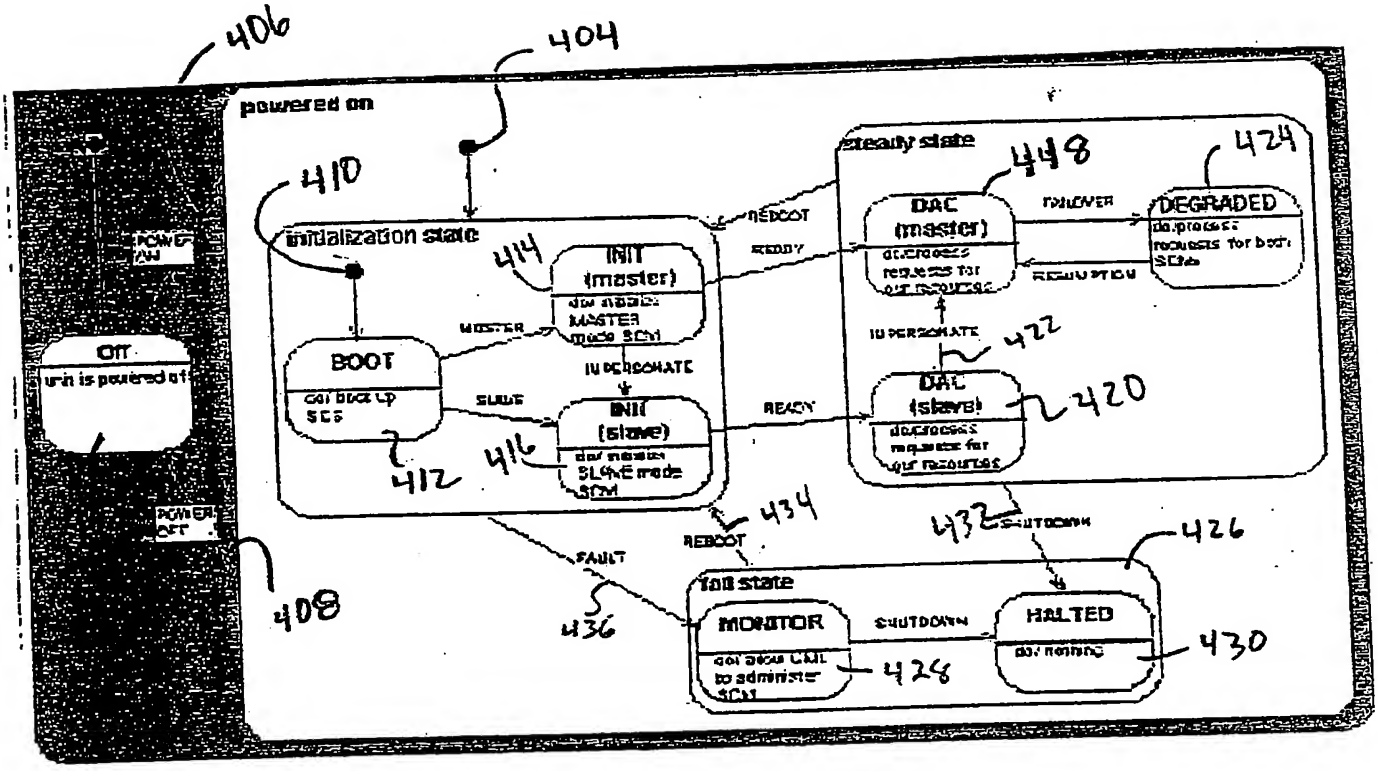
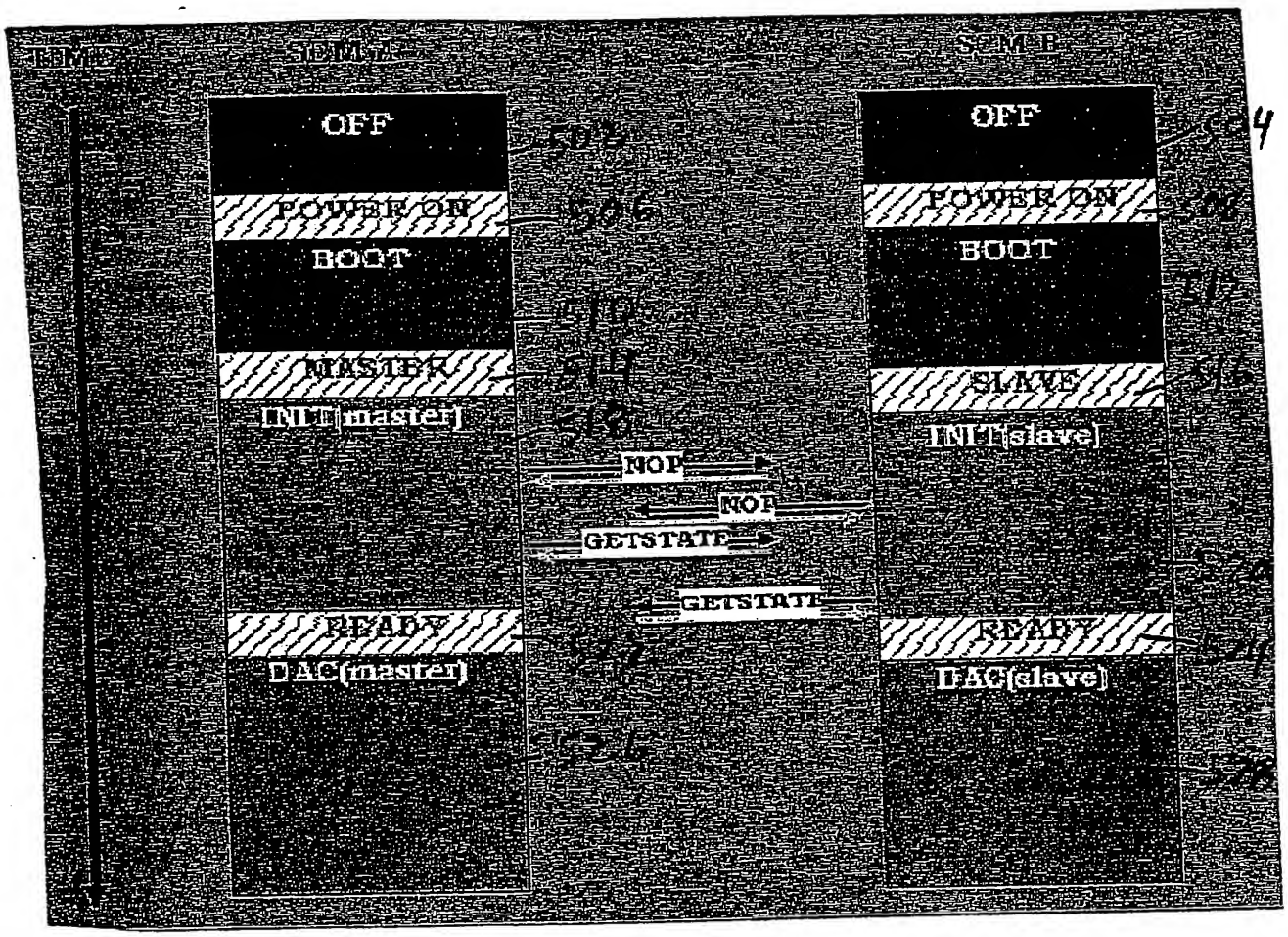

SCM A
SCM B
monitor
STATUS Generator
STATUS Monitor
control
interface
transition
hasm1.glk
110
300
316
320
322
324
326
328
330
332
334
312
314

FIG 3



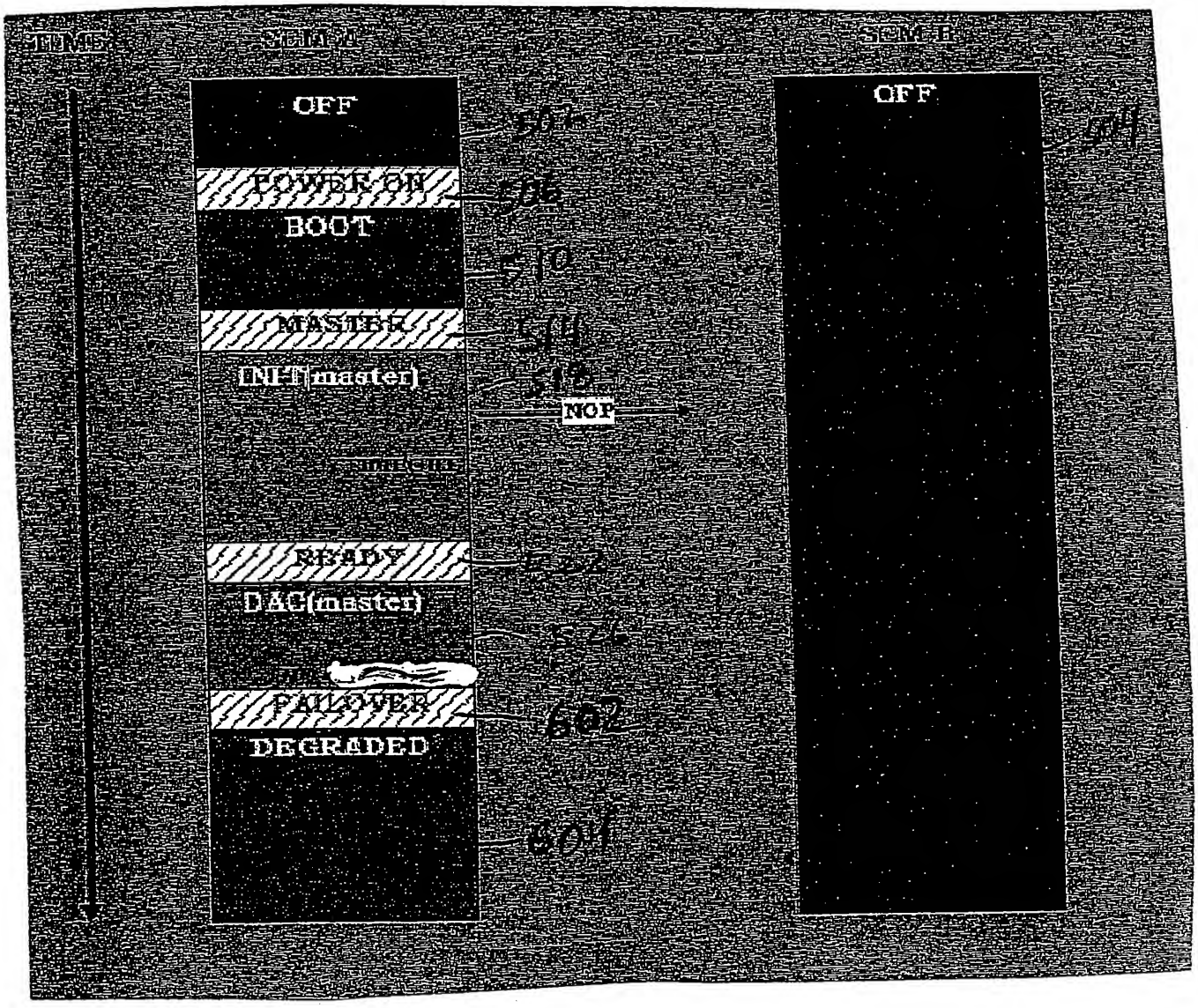
400

FIG. 4


SEMI
SEMI
OFF
POWER ON
BOOT
MASTER
INIT[master]
READY
DAC[master]
OFF
POWER ON
BOOT
SLAVE
INIT[slave]
READY
DAC[slave]
NOP
NOP
GETSTATE
GETSTATE

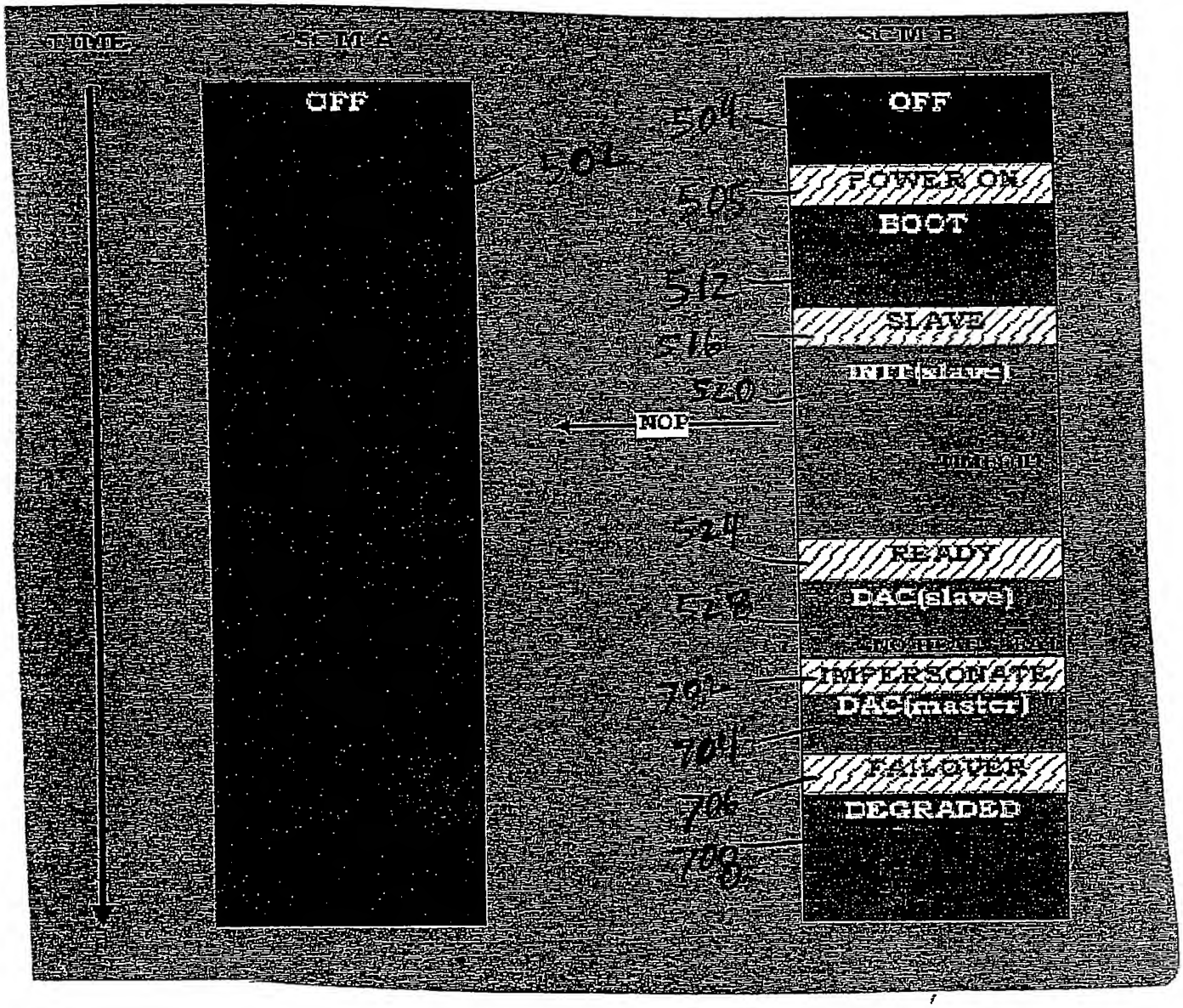
500

FIG. 5



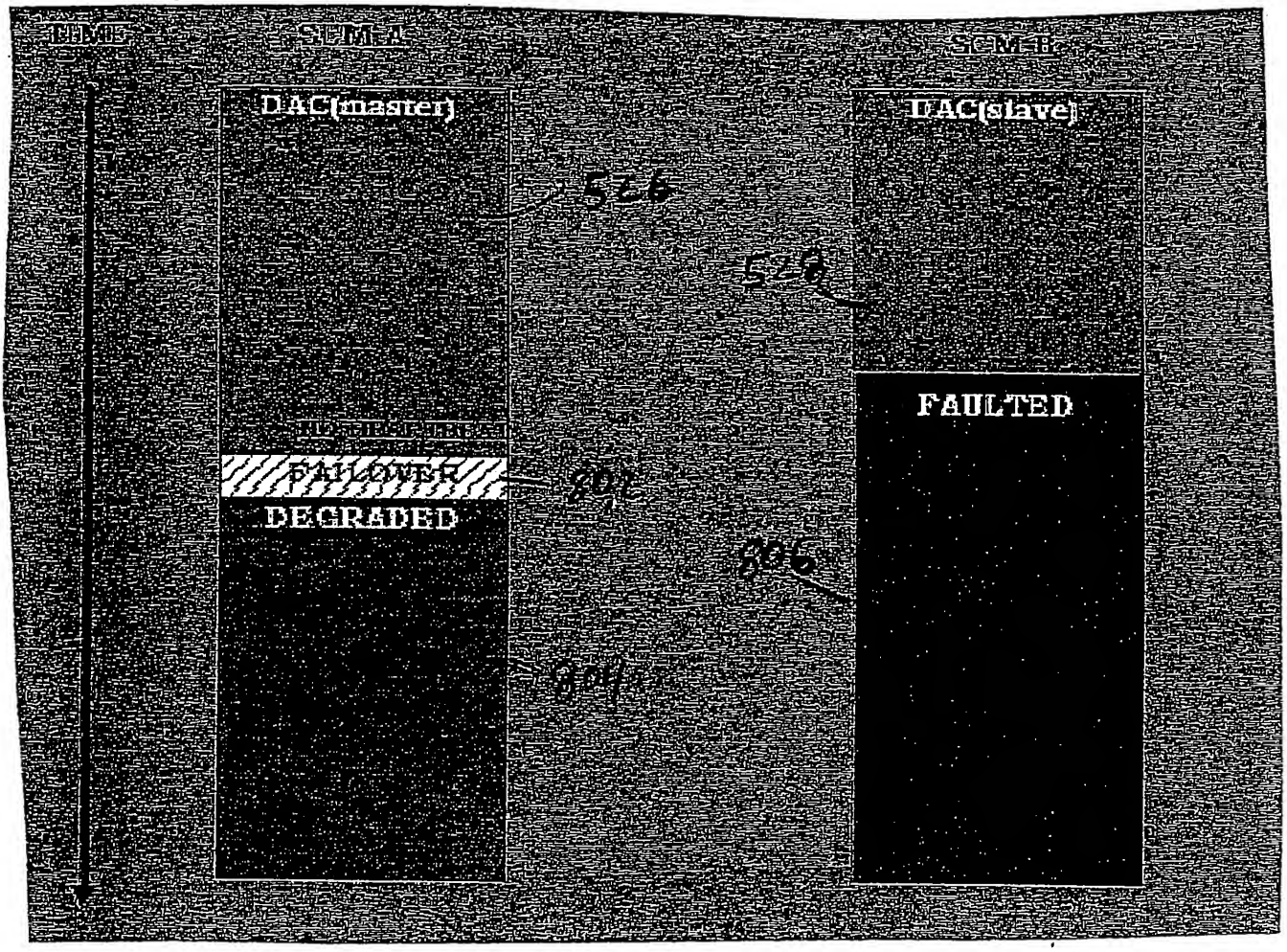
600

FIG. 6


TIME
SCM A
SCM B
OFF
OFF
504
502
POWER ON
505
BOOT
512
SLAVE
516
520
NOP
524
READY
526
DAC(slave)
IMPERSONATE
702
DAC(master)
704
FAILOVER
706
DEGRADED
708

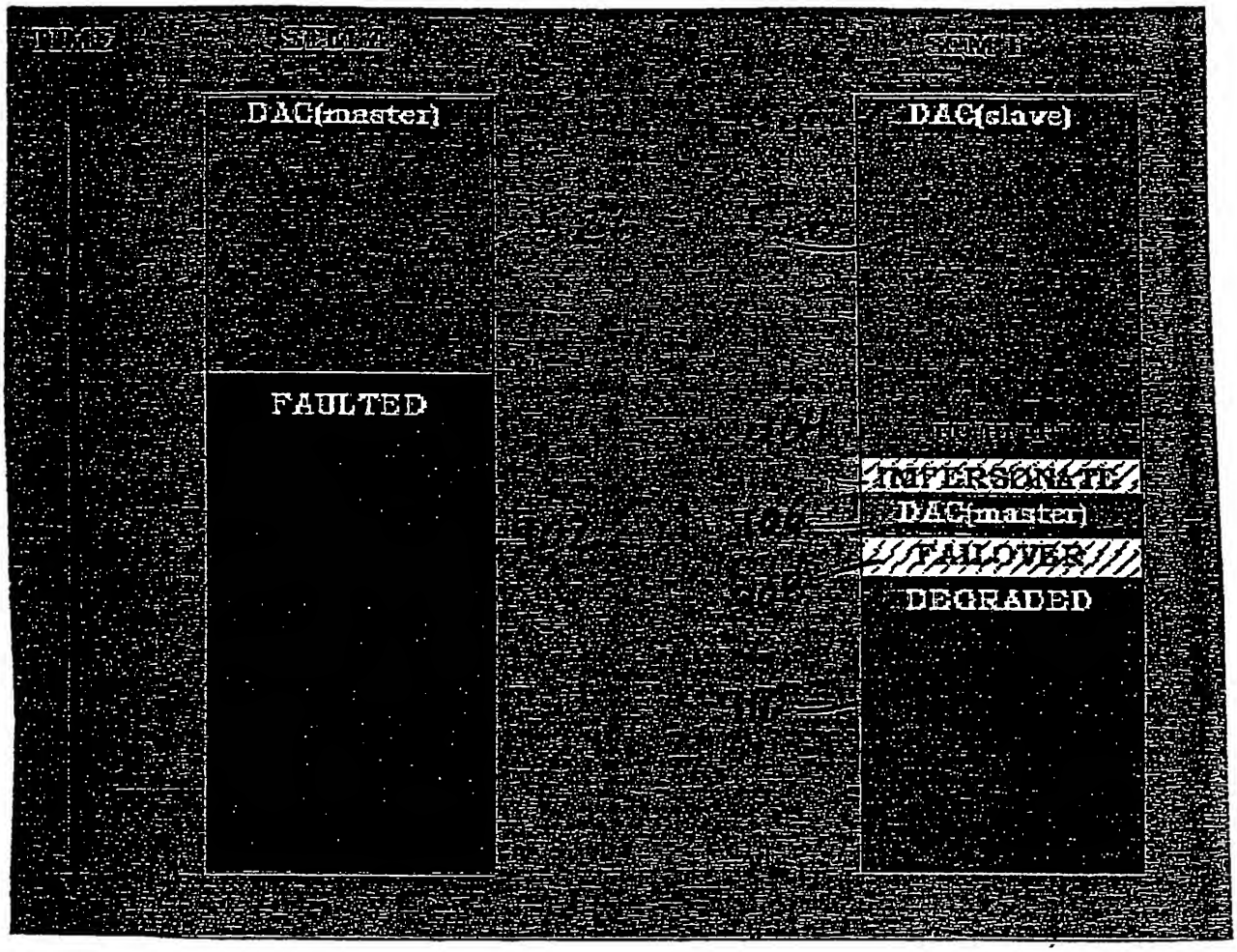
700

FIG. 7



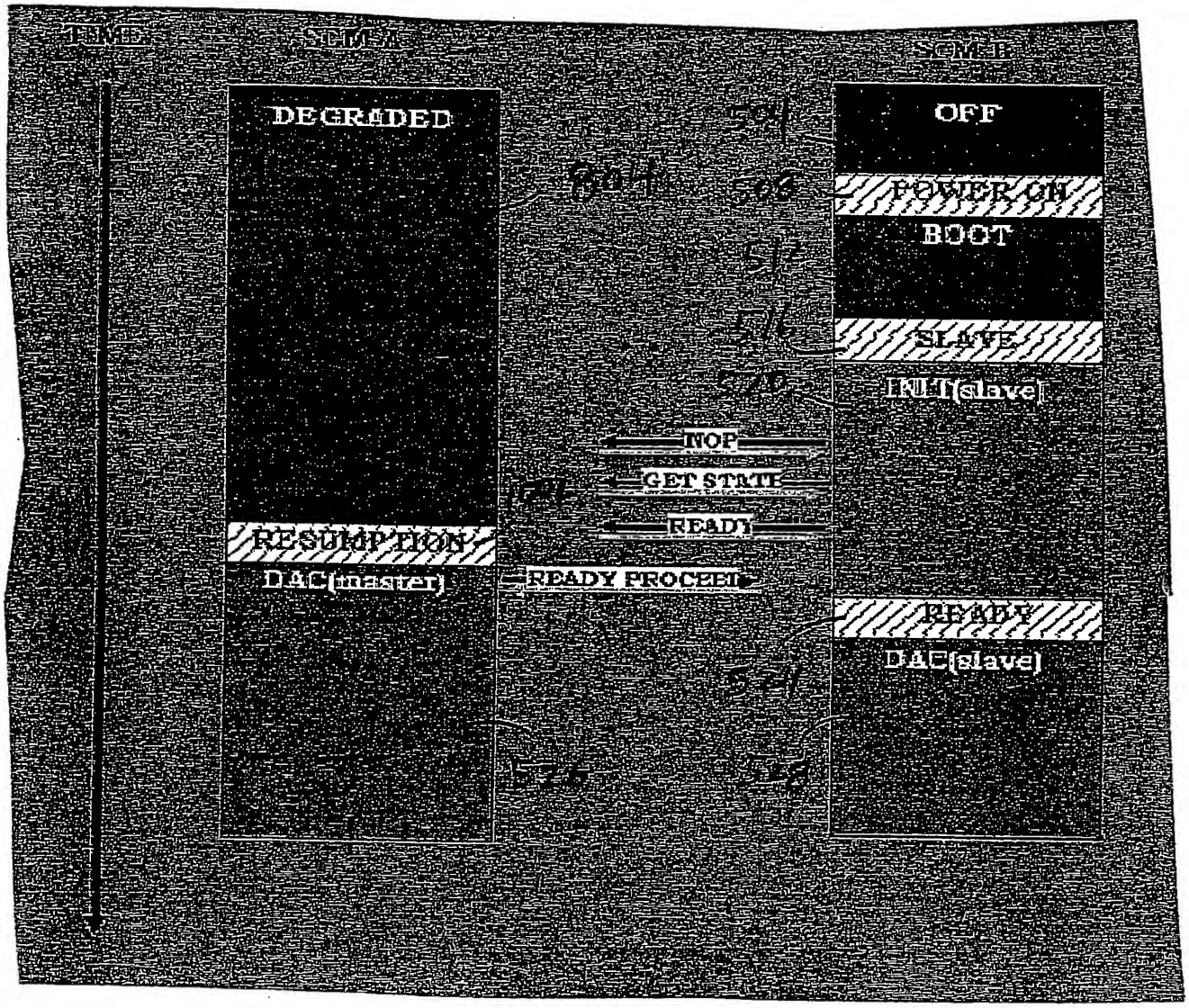
800

FIG 8


DAC(master)
FAULTED
DAC(slave)
IMPERSONATE
DAC(master)
FAILOVER
DEGRADED

900

FIG 9



1000

FIG. 10

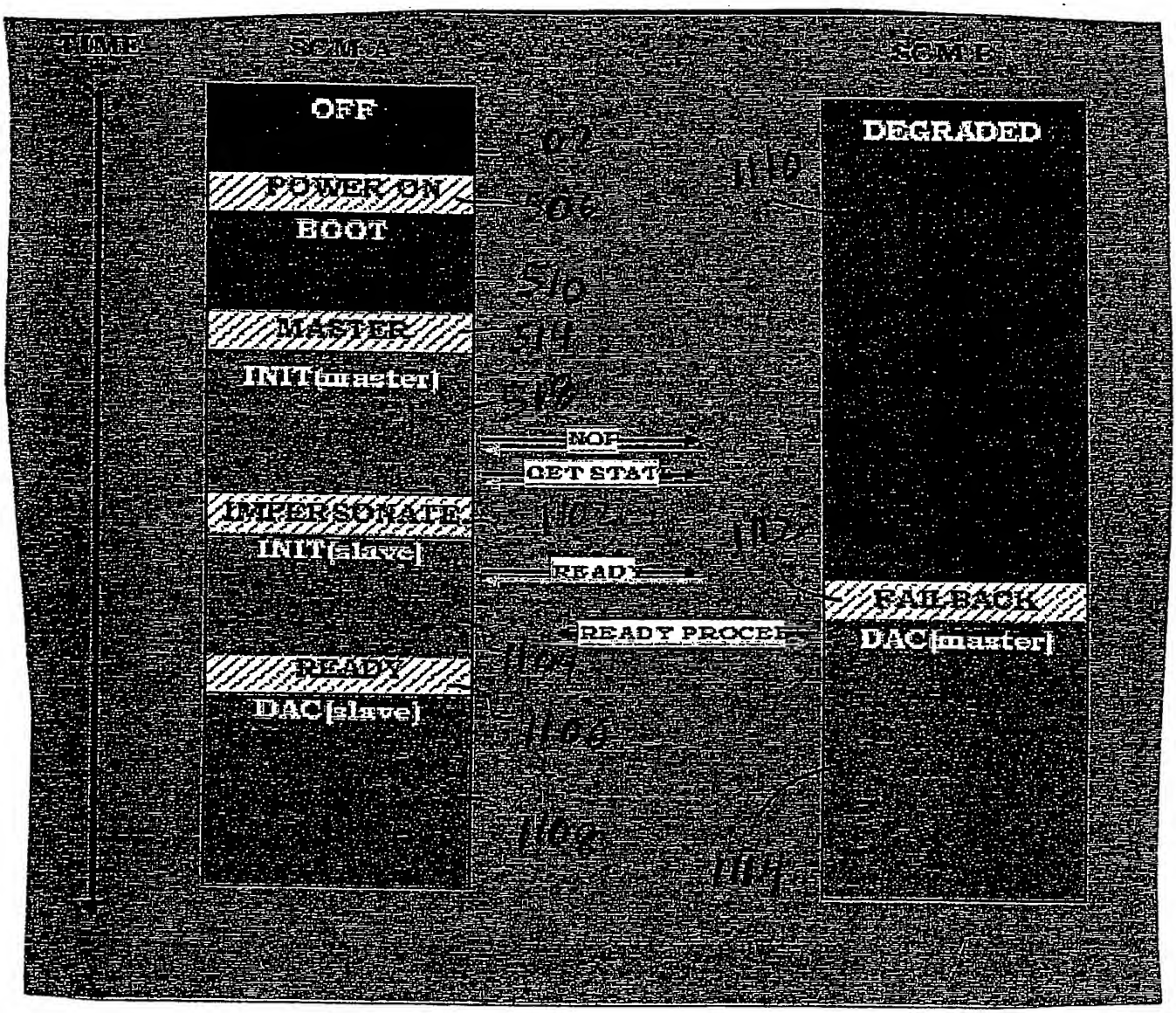

TIME
SCM A
SCM B
OFF
POWER ON
BOOT
MASTER
INIT[master]
IMPERSONATE
INIT[slave]
READY
DAC[slave]
DEGRADED
FAILBACK
DAC[master]
NOP
GET STAT
READY
READY PROCE
1102
1100

FIG 11

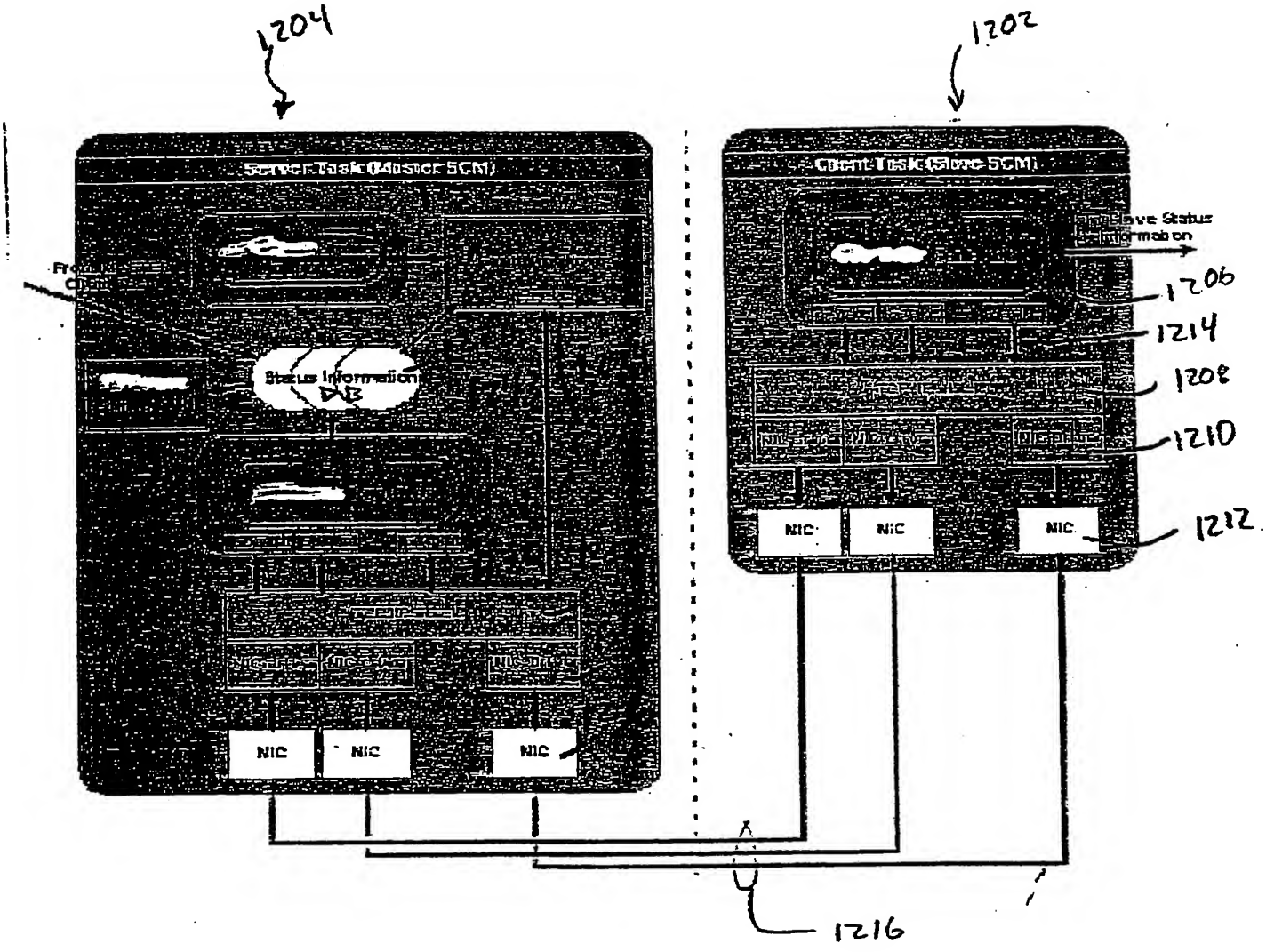


FIG. 12

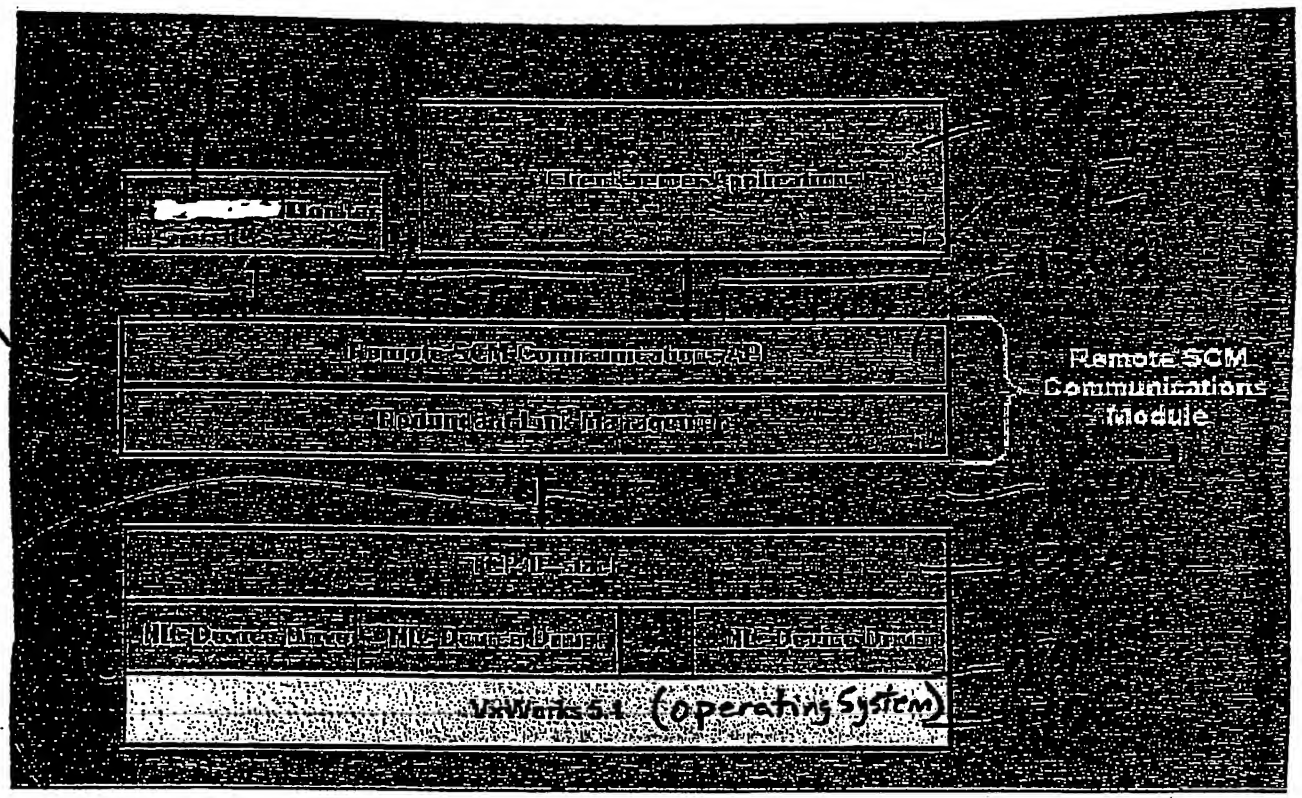

306
Remote SCM Communications Module
TCP/IP
VxWorks 5.1 (Operating System)

FIG. 13

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 November 2001 (01.11.2001)

PCT

(10) International Publication Number
WO 01/082080 A2

(51) International Patent Classification[7]: G06F 11/00

(21) International Application Number: PCT/US01/12889

(22) International Filing Date: 20 April 2001 (20.04.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/552,781 20 April 2000 (20.04.2000) US

(71) Applicant: CIPRICO, INC. [US/US]; Suite 60, 2800 Campus Drive, Plymouth, MN 55441 (US).

(72) Inventors: MCMILLAN, Ben, H., Jr.; 125 Marvin Road, Middletown, NJ 07748 (US). DAVIS, Daniel, A.; 33 S. First Avenue, Apartment 2, Highland Park, NJ 08904 (US).

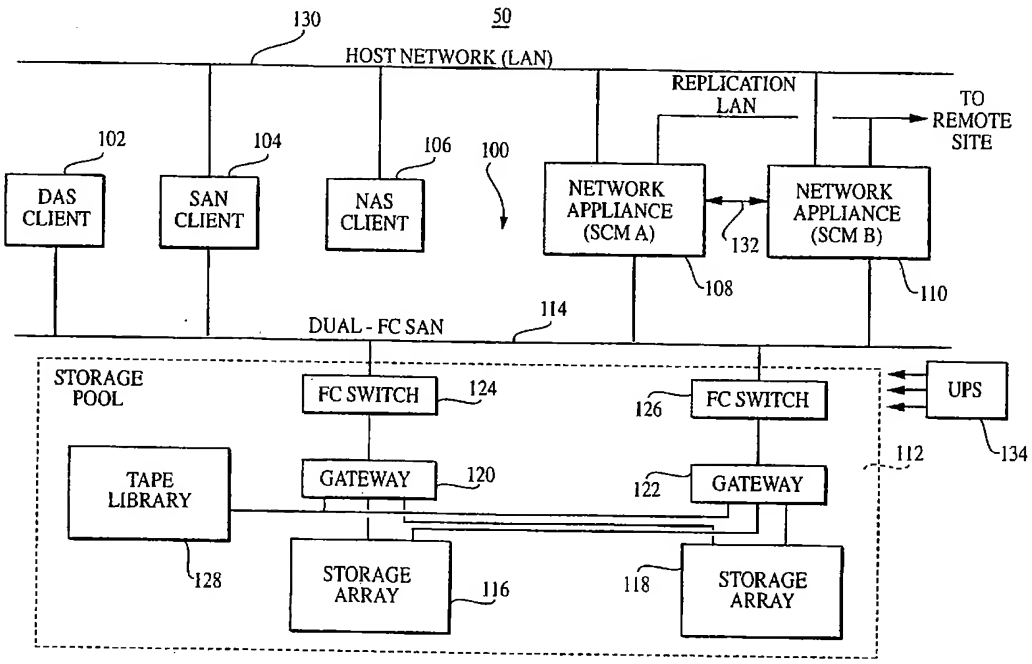
(74) Agent: MACMASTERS, Thomas, L.; Fredrikson & Byron, P.A., 1100 International Center, 900 Second Avenue South, Minneapolis, MN 55402 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: FAULT-TOLERANT, HIGH AVAILABILITY NETWORK APPLIANCE



(57) Abstract: A method and apparatus for performing fault-tolerant network computing. The apparatus comprises a pair of network appliances coupled to a network. The appliances interact with one another to detect a failure in one appliance and instantly transition operations from the failed appliance to a functional appliance.

WO 01/082080 A2

Published:
— *without international search report and to be republished upon receipt of that report*

(48) Date of publication of this corrected version:
8 August 2002

(15) Information about Correction:
see PCT Gazette No. 32/2002 of 8 August 2002, Section II

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

FAULT-TOLERANT, HIGH AVAILABILITY NETWORK APPLIANCE

BACKGROUND OF THE DISCLOSURE

1. Field of the Invention

The invention relates to network appliances and, more particularly, the invention relates to a method and apparatus for providing fault-tolerant, high availability network appliances.

2. Description of the Background Art

Data processing and storage systems that are connected to a network to perform task specific operations are known as network appliances. Network appliances may include a general purpose computer that executes particular software to perform a specific network task, such as file server services, domain name services, data storage services, and the like. Because these network appliances have become important to the day-to-day operation of a network, the appliances are generally required to be fault-tolerant. Typically, fault tolerance is accomplished by using redundant appliances, such that, if one appliance becomes disabled, another appliance takes over its duties on the network. However, the process for transferring operations from one appliance to another leads to a loss of network information. For instance, if a pair of redundant data storage units are operating on a network and one unit fails, the second unit needs to immediately perform the duties of the failed unit. However, the delay in transitioning from using one storage unit to another causes some data to not be stored and will be lost.

Therefore, a need exists in the art for fault-tolerant, highly available network appliances that

seamlessly transition from one appliance to another to provide redundant appliance operations.

SUMMARY OF THE INVENTION

The disadvantages associated with the prior art are overcome by the present invention of a method and apparatus for performing fault-tolerant network computing. The apparatus comprises a pair of network appliances coupled to a network. The appliances interact with one another to detect a failure in one appliance and instantly transition operations from the failed appliance to a functional appliance. Each appliance comprises shared configuration and state information such that the transition of services from a failed appliance to an operating appliance is seamless.

In one embodiment of the invention, the apparatus comprises a pair of storage controller modules (SCM) that are coupled to a storage pool, i.e., one or more data storage arrays. The storage controller modules are coupled to a host network (or local area network (LAN)). The network comprises a plurality of client computers that are interconnected by the network.

In operation, the client computers request access to the storage pool via a read or write request that is sent through the network to the SCMs. The storage control managers handle the request by routing the request to an appropriate storage array with the storage pool. If one of the storage controller modules were to fail, the other manager would instantly begin handling requests from the network that would otherwise be handled by the failed appliance. The SCMs share configuration and state information so that the operational SCM can rapidly take over the resources of the failed SCM.

BRIEF DESCRIPTION OF THE DRAWINGS

The teachings of the present invention can be readily understood by considering the following detailed description in conjunction with the accompanying drawings, in which:

FIG. 1 depicts a block diagram of one embodiment of the present invention;

FIG. 2 depicts a block diagram of a pair of storage controller modules;

FIG. 3 depicts a functional block diagram of the pair of storage controller modules;

FIG. 4 depicts a state diagram for a storage controller module;

FIG. 5 depicts a flow diagram of a normal boot process for the pair of storage controller modules;

FIG. 6 depicts a flow diagram of a boot process having a faulted slave storage controller module;

FIG. 7 depicts a flow diagram of a boot process having a faulted master storage controller module;

FIG. 8 depicts a flow diagram of the operation of the pair of storage controller modules when the slave storage controller module fails;

FIG. 9 depicts a flow diagram of the operation of the pair of storage controller modules when the master storage controller module fails;

FIG. 10 depicts a flow diagram of the operation of the pair of storage controller modules when the slave storage controller module resumes operation;

FIG. 11 depicts a flow diagram of the operation of the pair of storage controller module when the master storage controller module performs a failback operation.

FIG. 12 depicts a functional block diagram of the status monitoring system of the pair of storage controller modules; and

FIG. 13 depicts the software architecture for a storage controller module.

To facilitate understanding, identical reference numerals have been used, where possible, to designate identical elements that are common to the figures.

DETAILED DESCRIPTION

One embodiment of the invention is a modular, high-performance, highly scalable, highly available, fault tolerant network appliance that is illustratively embodied in a data storage system. FIG. 1 depicts a data processing system 50 comprising a plurality of client computers 102, 104, and 106, a host network 130, and a storage system 100. The storage system 100 comprises a plurality of network appliances 108 and 110 and a storage pool 112. The plurality of clients comprise one or more of a network attached storage (NAS) client 102, a direct attached storage (DAS) client 104 and a storage area network (SAN) client 106. The plurality of network appliances 108 and 110 comprise a storage controller module A (SCM A) 108 and storage controller module B (SCM B) 110. The storage pool 112 is coupled to the storage controller modules 108, 110 via a fiber channel network 114. One embodiment of the storage pool 112 comprises a pair of storage arrays 116, 118 that are coupled to the fiber channel network 114 via a pair of fiber channel switches 124, 126 and a communications gateway 120, 122. A tape library 128 is also provided for storage backup.

In storage system 100, the DAS client directly accesses the storage pool 112 via the fiber channel network 114, while the SAN client accesses the storage pool 112 via both the LAN 130 and the fiber channel network 114. For example, the SAN client 104 communicates via the LAN with the SCMs 108, 110 to request access to the storage pool

112. The SCMs inform the SAN client 104 where in the storage arrays the requested data is located or where the data from the SAN client is to be stored. The SAN client 104 then directly accesses a storage array using the location information provided by the SCMs. The NAS client 106 only communicates with the storage pool 112 via the SCMs 108, 110. Although a fiber channel network is depicted as one way of connecting the SCMs 108, 110 to the storage pool 112, the connection may be accomplished using any form of data network protocol such as SCSI, HIPPI and the like.

Dependability is one of the major design goals of the storage system 100. Dependability is addressed in four areas, reliability, availability, safety, and security. Through the use of hardware fault tolerance and the deployment of redundant components, the aim of the invention is to eliminate any single points of failure. The availability of the storage system is addressed through a high availability software module (HASM) that provides a high availability environment in which storage applications can operate. This high availability or software fault tolerance aspect is addressed in four sections, fault elimination, fault removal, fault tolerance and fault avoidance techniques. The safety aspects of the system protects the data stored in the storage system so that no data corruption occurs. This is addressed through the use of a redundant array of inexpensive disk (RAID) technique of storing data in the storage arrays. The last aspect of dependability is security that is implemented in the file and directory protection that allows users to protect their data from unauthorized access.

The modularity of the storage system 100 allows a wide range of markets to be addressed by a single product family. This is due to the fact that this storage system can be customized to a customer's needs simply by choosing

the appropriate components. This modularity also allows the customer to grow the storage capabilities as needed. The scalability of the storage system allows a customer to start with a relatively in-expensive single SCM and one storage array configuration, then add additional SCMs and storage arrays to grow the system into an enormous storage system.

The storage system 100 provides substantial flexibility in terms of connectivity configurations. The use of common PCI I/O cards provide connections to direct channel, networks, and SANs. New connectivity options are quickly adopted since only the I/O card and software drivers need be developed and upgraded. The rest of the storage system need not be changed to facilitate implementation of system improvements.

The storage system is a hierarchy of system components that are connected together within the framework established by the system architecture. The major active system level components are:

SCM - Storage Controller Module
SDM - Storage Device Module (Storage Pool)
UPS - Uninterruptible Power Supply
Fibre channel switches, hubs, routers and gateways

The system architecture provides an environment in which each of the storage components that comprise the storage system embodiment of the invention operate and interact to form a cohesive storage system.

The architecture is centered around a pair of SCMs 108 and 110 that provide storage management functions. The SCMs are connected to a host network that allows the network community to access the services offered by the SCMs 108, 110. Each SCM 108, 110 is connected to the same

set of networks. This allows one SCM to provide the services of the other SCM in the event that one of the SCMs becomes faulty. Each SCM 108, 110 has access to the entire storage pool 112. The storage pool is logically divided by assigning a particular storage device (array 116 or 118) to one of the SCMs 108, 110. A storage device 116 or 118 is only assigned to one SCM 108 or 110 at a time. Since both SCMs 108, 110 are connected to the entirety of the storage pool 112, the storage devices 116, 118 assigned to a faulted SCM can be accessed by the remaining SCM to provide its services to the network community on behalf of the faulted SCM. The SCMs communicate with one another via the host networks. Since each SCM 108, 110 is connected to the same set of physical networks as the other, they are able to communicate with each other over these same links. These links allow the SCMs to exchange configuration information with each other and synchronize their operation.

The host network 130 is the medium through which the storage system communicates with the clients 104 and 106. The SCMs 108, 110 provide network services such as NFS and HTTP to the clients 104, 106 that reside on the host network 130. The host network 130 runs network protocols through which the various services are offered. These may include TCP/IP, UDP/IP, ARP, SNMP, NFS, CIFS, HTTP, NDMP, and the like.

From an SCM point of view, its front-end interfaces are network ports running file protocols. The back-end interface of each SCM provides channel ports running raw block access protocols.

The SCMs 108, 110 accept network requests from the various clients and process them according to the command issued. The main function of the SCM is to act as a network-attached storage (NAS) device. It therefore communicates with the clients using file protocols such as

NFSv2, NFSv3, SMB/CIFS, and HTTP. The SCM converts these file protocol requests into logical block requests suitable for use by a direct-attach storage device.

The storage array on the back-end is a direct-attach disk array controller with RAID and caching technologies. The storage array accepts the logical block requests issued to a logical volume set and converts it into a set of member disk requests suitable for a disk drive.

The redundant SCMs will both be connected to the same set of networks. This allows either of the SCMs to respond to the IP address of the other SCM in the event of failure of one of the SCMs. The SCMs support 10BaseT, 100BaseT, and Gigabit Ethernet. The SCMs can communicate with each other through a dedicated inter-SCM network 132 as a primary means of inter-SCM communications. This dedicated connection can employ 100BaseT Ethernet, Gigabit Ethernet or fibre channel. In the event of the failure of this link 132, the host network 130 may be used as a backup network. The SCMs 108, 110 connect to the storage arrays 116, 118 through parallel differential SCSI (not shown) or a fiber channel network 114. Each SCM 108, 110 may be connected through their own private SCSI connection to one of the ports on the storage array.

The storage arrays 116, 118 provide a high availability mechanism for RAID management. Each of the storage arrays provides a logical volume view of the storage to a respective SCM. The SCM does not have to perform any volume management.

The UPS 134 provides a temporary secondary source of AC power source in the event the primary source fails. This allows time for the storage arrays 116, 118 to flush the write-back cache and for the SCMs 108, 110 to perform an orderly shutdown of network services. The UPS is monitored by the SCMS through the serial port or over the host network using SNMP.

FIG. 2 depicts an embodiment of the invention having the SCMs 108, 110 coupled to the storage arrays 116, 118 via SCSI connections 200. Each storage array 116, 118 comprises an array controller 202, 204 coupled to a disk enclosure 206, 208. The array controllers 202, 204 support RAID techniques to facilitate redundant, fault tolerant storage of data. The SCMs 108, 110 are connected to both the host network 130 and to array controllers 202, 204. Note that every host network interface card (NIC) 210 connections on one SCM is duplicated on the other. This allows a SCM to assume the IP address of the other on every network in the event of a SCM failure. One of the NICs 212 in each SCM 108, 110 is dedicated for communications between the two SCMs.

On the target channel side of the SCM, note that each SCM 108, 110 is connected to an array controller 202, 204 through its own host SCSI port 214. All volumes in each of the storage arrays 202, 204 are dual-ported through SCSI ports 216 so that access to any volume is available to both SCMs 108, 110.

Storage Controller Module (SCM) Hardware

The SCM 108, 110 is based on a general purpose computer (PC) such as a ProLiant 1850R manufactured by COMPAQ Computer Corporation. This product is a Pentium PC platform mounted in a 3U 19" rack-mount enclosure. The SCM comprises a plurality of network interface controls 210, 212, a central processing unit (CPU) 218, a memory unit 220, support circuits 222 and SCSI parts 214. Communication amongst the SCM components is supported by a PCI bus 224. The SCM employs, as a support circuit 222, dual hot-pluggable power supplies with separate AC power connections and contains three fans. (One fan resides in each of the two power supplies). The SCM is, for example,

based on the Pentium III architecture running at 600 MHz and beyond. The PC has 4 horizontal mount 32-bit 33 MHz PCI slots. As part of the memory (MEM) unit 220, the PC comes equipped with 128 MB of 100 MHz SDRAM standard and is upgradable to 1 GB. A Symbios 53c8xx series chipset resides on the 1850R motherboard that can be used to access the boot drive.

The SCM boots off the internal hard drive (also part of the memory unit 220). The internal drive is, for example, a SCSI drive and provides at least 1 GB of storage. The internal boot device must be able to hold the SCSI executable image, a mountable file system with all the configuration files, HTML documentation, and the storage administration application. This information may consume anywhere from 20 to 50 MB of disk space.

In a redundant SCM configuration, the SCM's 108, 110 are identically equipped in at least the external interfaces and the connections to external storage. The memory configuration should also be identical. Temporary differences in configuration can be tolerated provided that the SCM with the greater number of external interfaces is not configured to use them. This exception is permitted since it allows the user to upgrade the storage system without having to shut down the system. As mentioned previously, one network port can be designated as the dedicated inter-SCM network. Only SCMs and UPS's are allowed on this network 132.

Storage Device Module (SDM) Hardware

The storage device module (storage pool 112) is an enclosure containing the storage arrays 116 and 118 and provides an environment in which they operate.

One example of a disk array 116, 118 that can be used with the embodiment of the present invention is the

Synchronix 2000 manufactured by ECCS, Inc. of Tinton Falls, New Jersey. The Synchronix 2000 provides disk storage, volume management and RAID capability. These functions may also be provided by the SCM through the use of custom PCI I/O cards.

Depending on the I/O card configuration, multiple Sychronix 2000 units can be employed in this storage system. In one illustrative implementation of the invention, each of the storage arrays 116, 118 uses 4 PCI slots in a 1 host/3 target configuration, 6 SCSI target channels are available allowing six Synchronix 2000 units each with thirty 50GB disk drives. As such, the 180 drives provide 9 TB of total storage. Each storage array 116, 118 can utilize RAID techniques through a RAID processor 226 such that data redundancy and disk drive fault tolerance is achieved.

SCM Software

Each SCM 108,110 executes a high availability software module (HASM), which is a clustering middleware that provides the storage system 100 with a high availability environment. The HASM is a collection of routines and subroutines stored in the memory units of each SCM that, when executed, provides the functionality of the present invention. The HASM allows two SCMs to run in a dual-active symmetrical configuration meaning that either SCM can take over for the other in the event of a SCM fault. The SCM hardware fault tolerance allows a redundant SCM to manage the resources and present the services to the network community of a SCM that has faulted. The software fault tolerance allows all state-less and state-full NAS applications and services such as NFS, HTTP, and CIFS to run in a high availability environment. The software fault tolerance provides a mechanism to detect and recover from

faults in the software whether they evolved from a hardware error or a software design or implementation error. This software allows the SCM to assume the resources and services of a faulted SCM in as transparent a manner as possible within the capabilities of the existing protocols. Clients logged into the faulted SCM must continue to believe they are logged into the same SCM.

The HASM monitors the state of both the local SCM and the remote SCM. The local SCM is continually monitored through the local health monitor (LHM) as well as by the individual applications running on the SCM. If an application finds that a critical error has occurred that prevents the SCM from operating correctly, this application can force the SCM to reboot thus transferring control to the remote SCM. If the LHM has determined that the system is not operating correctly, a surrender to the remote SCM can take place so that the local SCM can reboot and hopefully correct the situation. Attempts to detect internal error conditions must occur as soon as possible. It is the intent of this design to detect and initiate recovery on the order of a few seconds or less. The remote SCM is monitored using status messages that are exchanged between the two SCMs. In one embodiment of the invention, status messages are transmitted across all available network channels. In the event a number of network channels fail, a false detection of a faulted SCM will not occur. The remote SCM is not considered faulted unless it either indicates it has faulted or all status message channels fail. The time-out value is tunable to maximize failover time yet minimize false alarms. If a failover operation is initiated, the procedure must be completed as fast as possible. The primary source of delay in failing over is the checking of the integrity of the file systems (fsck). It is anticipated that this should not exceed a few minutes except in the most strenuous of cases.

If using the host networks as a primary conduit for status message communications is unacceptable, the serial ports may be used in conjunction with the networks to provide a backup for the networks in case the entire host network fails. This would prevent one of the SCMs from thinking the other has failed when in fact this has not occurred.

FIG. 3 is a functional block diagram of the HASM 300 of each of the SCMs. The HASM 300 consists of several major components, each of which work together to create a high availability environment in which embedded NAS applications can run. These components are classified as control, monitor, service, and transition modules. These components include the high availability software controller (HASC) 302, the status monitor (SM) 304, the local health monitor (LHM) 306, the remote SCM communications link (RSCM) 308, the persistent shared object module (PSOM) 310, the shared file manager (SFM) 312, the transition functions 314 (FAILBACK, FAILOVER, FAULT, IMPERSONATE, MASTER, RESET, SHUTDOWN, and SLAVE) and the configuration transaction control module (CTCM) 316. These modules have the following responsibilities:

HASC 302 - High Availability Software Controller - controls the HASM by gathering information from the SM, LHM, and RSCM to determine the state of the SCM if a transition to a new state of operation is required.

SM 304 - Status Monitor - monitors and assesses the health of the remote SCM.

LHM 306 - Local Health Monitor - monitors and assess the health of the local SCM. Has the ability to restart services tasks that have terminated due to nonrecoverable errors.

RSCM 308 - Remote SCM Communications Manager - provides a reliable redundant communications link between the two SCMs for purposes of information exchange and synchronization. This provides the platform over which all inter-SCM communications take place. The RSCM is described in detail in U.S. patent application serial number _____ filed simultaneously herewith (Attorney docket ECCS 007), which is incorporated herein by reference.

PSOM 310 - Persistent Shared Object Manager - provides an object paradigm that allows objects to be distributed between the two SCMs. This module guarantees persistence of objects across power cycles.

SFM 312- Shared File Manager - provides a mechanism for keeping configuration files synchronized between the two SCMs.

CTCM 316 - Configuration Transaction Control Module - provides a configuration transaction paradigm to mark configuration changes to determine configuration state on power up.

FAILBACK module 318 - transitions the SCM from the DEGRADED state to the DAC(master) state.

FAILOVER module 320 - transitions the SCM from the DAC(master) state to the DEGRADED state.

FAULT module 322 - transitions the SCM into the MONITOR mode where configuration corrections and analysis can be made.

IMPERSONATE module 324 - allows a SCM to impersonate the other SCM when the MASTER faults (the SLAVE impersonates the MASTER) and when the MASTER boots and discovers the SLAVE is running in DEGRADED mode (the MASTER impersonates the SLAVE)

MASTER module 326 - transition the SCM from BOOT state to INIT(master) state.

READY module 328 - transitions the SCM from INIT state to DAC state for both the MASTER and SLAVE SCMs.

RESET module 330 - transitions the SCM from the current operational state to the BOOT state.

SHUTDOWN module (included in the RESET module 330)- transition the SCM from the current operational state to the HALTED state.

SLAVE module 332 - transitions the SCM from the BOOT state to the INIT(slave) state.

The HASM 300 allows stateless applications to perform in a high availability manner without modification. Maintenance of any configuration files is handled by the HASM 300 through the system administration module (SAM) (not shown). The SAM will invoke the HASM 300 to keep configuration files synchronized between the two SCMs 108, 110. In fact, the SAM is run as a stateful high availability service on top of the HASM 300. In the case of stateful applications such as the SAM, a set of API calls are provided that allow the state information to be synchronized between the two SCMs. This is provided primarily by the RSCM 308, PSOM 310, SFM 312, and the CTCM 316 as described in detail below with respect to FIG. 13.

The HAS controller (HASC) 302 is the central control module for the HASM. The HASM 300 gathers its information from monitoring and service modules, invokes transition modules to perform transitions from state to state, and communications with the remote HASC to synchronize and invoke operations on the remote SCM. The HASC 302 determines the state of the HASM 300 based on the information it gathers. HASC 302 is responsible for maintaining the current state of the SCM.

The monitoring modules are SM 304 and LHM 306. The SM 304 contains a status generator and status monitor. The status generator generates status messages that are transmitted to the remote SCM. The status monitor is responsible for listening to the status messages and gathering information based on their timely arrivals and or failures to be received at all. One particular technique that can be used to communicate and monitor status information is a heartbeat signal technique that is described in U.S. patent application serial number _____ filed simultaneously herewith (Attorney docket ECCS 006, which is incorporated herein by reference.

The LHM 306 monitors the local SCM looking for discrepancies in its operation. If either of these two modules 304, 306 reports a failure to the HASC 302, the HASC 302 performs the appropriate transition based on what faulted.

The HASC also takes input from the CTCM 316 and RSCM 308. The CTCM 316 is used only during initialization by the HASC 302 to determine the initial state of the HASM 300 software. The RSCM 308 provides error status of the various network channels to determine if the remote SCM is responding to requests properly.

The service modules are the RSCM 308, CTCM 316, PSOM 310, and SFM 312. These modules provide services that are used by components of the HASM 300 and high available

stateful applications. They also provide some feedback on the operation of the remote SCM which is used in conjunction with the SM's information in evaluating the condition of the remote SCM.

The transition modules 318 through 332 provide a capability of transitioning the SCM and storage system from one state to another and are responsible for making storage system transitions from DEGRADED mode to DUAL-ACTIVE CONFIGURATION mode and back to DEGRADED mode as well as for making SCM transitions from MASTER to SLAVE or from SLAVE to MASTER.

In order for the HASM 300 to function correctly, some requirements must be met regarding the functionality of the system in which the HASM 300 is to run:

1. IP aliasing - The network interfaces must be able to respond to multiple IP addresses.
2. Gratuitous ARP notification - This message must be broadcast to notify other computers on the host networks that the MAC addresses has changed for a particular IP address.
3. Initiator ID - The HASM must be able to control the initiator ID of the SCSI HBAs that reside in the SCM. The first SCSI HBA is always designated for exclusive use of the local SCM. Since this is the case, the initiator ID of this SCSI HBA may be set to 7 by default. The remaining SCSI HBAs are used to shared storage on the back-end of the storage system. Due to SCSI requirements, two devices can not have the same SCSI ID on a SCSI bus. Therefore, one of the SCMs must be configured to use an initiator ID of 7 and the other, an initiator ID of 6.
4. All NICs on one SCM must be connected to the NICs on the other SCM.

5. Both SCMs must have access to the same storage pool.
6. The File Systems must be logged structure with checkpointing.
7. File System write caching must be disabled. This prevents cached data from being lost during a SCM failure.
8. All applications running in the HA environment that do not use the HASM services, must be persistent across restarts.
9. Interface and device errors must be reportable to the HASM through a callback function interface or some other similar mechanism.

To achieve seamless transition between SCMs upon the occurrence of a fault, stateful applications require state information to be consistent between SCMs. This allows the remote SCM to have up-to-date information regarding the state of execution of the local SCM. In case, the local SCM faults, the remote SCM has the required information to continue on where the local SCM left off. The requirements of this state information depends upon the specified application. The HASM does provide a set of services that allows state information to be propagated to the other SCM. The stateful applications that require such services are the HTTP server, the NFS server, and the CIFS server. Additionally, the HASM may employ the use of alternate hardware channels to provide low latency state coherency operations.

As mentioned earlier, the system administration module is a stateful application which will use the services of the HASM to maintain state coherency. The SAM uses the services of the RSCM, SFM, CTCM, and PSOM to achieve state coherency. Only the SAM running on the MASTER SCM is allowed to administer changes to the configuration.

Master/Slave Mode Operation of a SCM

A SCM operates in either MASTER mode or SLAVE mode. The SCM running in MASTER mode is the highest authority in the storage system. There can be only one SCM running in MASTER mode. This designation is for internal purposes only and is transparent to the user of the storage system. The operator (system administration) is the only user who is aware of the MASTER and SLAVE mode roles each of the SCMs play. The first SCM configured into the storage system is always designated as the MASTER mode SCM. Any additional SCMs are designated as SLAVE mode SCMs. There may be one or more SLAVE mode SCMs.

Upon power up, a properly running redundant SCM configuration runs in their configured operational mode (MASTER or SLAVE). The SCM configured for MASTER mode runs in MASTER mode and the SCM configured for SLAVE mode runs in SLAVE mode. If the MASTER mode SCM fails, the SLAVE mode SCM must transition itself into the MASTER mode state. The configured SLAVE mode SCM will continue to run in MASTER mode until it is powered down or faults. The configured MASTER mode SCM upon reboot, will configured itself to run in SLAVE mode until it is powered down or the remote SCM faults.

The applications that execute in the HASM will assume different roles depending on if the SCM on which it resides in executing in MASTER or SLAVE mode. Not all applications behave in this manner. For example, the status monitor runs identically regardless of which mode the SCM is running in. The name_OpMode() function is used to notify the software module as to which mode the SCM is operating in and when a transition from one mode to another occurs.

The storage administration module (SAM) only allows administration on the MASTER mode SCM. All administration commands from the network community must be directed to the

MASTER mode SCM. A request to the SLAVE SCM will return a redirection request to the MASTER SCM.

High Availability Software Controller (HASC)

The HASC is responsible for determining the state of operation of the SCM and performing state transitions based on the information gathered by the SM, the LHM, and the CTCM. The SM and RSCM provide the HASC with status information about the remote SCM, the LHM provides information regarding the health of the local SCM, and the CTCM returns information regarding the state of the configuration information on initial powerup. The HASC gathers information from these sources to determine the correct operational state of the SCM. If the HASC determines that a state transition is required, it will invoke the appropriate routines to correctly perform the transition. For example, if the SCM is running in DAC mode and the SM reports back that the remote SCM is no longer generating or responding to status messages, the HASC will call the FM to perform a failover procedure. The HASC will transition the SCM from DAC mode to degraded mode operation. The HASC then invokes the name_OpMode() functions for all affected software modules to transition the operation of the SCM from DAC mode to degraded mode.

State and Transitions

An SCM runs in context of one of several different states. At the highest operating level, the SCM is either powered on or powered off. If powered on, the SCM is in one of three major state, initializing state, steady state, or fail state. The valid initialization states are BOOT and INIT. The valid fail states are HALTED, and MONITOR. The valid steady states are DAC and DEGRADED.

The SCM may also transition from state to state using the following transitions: POWER ON, READY, FAILOVER, FAILBACK, IMPERSONATE, SHUTDOWN, REBOOT, and POWER OFF.

On the order in which the various software modules are invoked to transition the SCM from state to state, it is recommended that all HASM specified modules be called prior to the software applications running in the HASM environment. This will facilitate the modularization of the HASM software such that it can be more easily ported to other hardware platforms in the future.

These opmodes, states and transitions are explained below:

SCM OpModes:

MASTER - the SCM is operating in the role as MASTER. It is the highest authority SCM in the storage system. This indicates that this SCM is the point of administration and control for the storage system. Only the MASTER controller can change the configuration. The EMM monitors the storage pool, the event log controls the storage of events, the PSOM controls the locking of persistent shared objects.

SLAVE - the SCM is operating in the role as SLAVE. This means that this SCM must follow the commands of the MASTER SCM.

SCM States:

POWERED OFF - the SCM is not powered on. It performs no operation in this state. It transitions to this state when powered is removed from the SCM. Its transitions out of this state to the INIT state after the power is applied to the SCM.

BOOT - [initialization state] - the SCM runs diagnostics and boots up the SCM off the internal hard drive of the SCM. While the system is operating in this state, name_Init() and name_Test() calls are made to initialization and test the various software modules that comprise the set of storage applications that interacts with the HASM. The SCM then determines its operational mode and transitions to the INIT state through either the MASTER or SLAVE transition.

INIT(master/slave) - [initialization state] - the SCM enters this mode for initialization purposes. This state allows the SCM to discover the remote SCM, establish contact with it to determine what its next course of action will be and join with it to create a highly available storage system. Upon completion of this step, the SCM will enter steady state in DAC mode. If the remote SCM is in DEGRADED mode and the local SCM is configured to run in MASTER mode, the local SCM must first transition to SLAVE mode and rerun its initialization process. If the remote SCM is in DEGRADED mode and local SCM is running in SLAVE mode (whether configured or current), the local SCM will remain in this step until the remote SCM has transitioned to DAC mode. The local SCM will then complete its initialization by initializing its resources and offering its assigned services to the network community.

DAC(master/slave) (Dual-Active Configuration) - [steady state] - this is the normal mode of operation of the SCM. This indicates that both SCMs are managing there assigned resources and are offering the services for which they were each configured. An SCM in this

state continually monitors the remote SCM. An SCM will run differently in this state depending on whether it is running in the MASTER role or the SLAVE role. When entering steady state for the first time, a SCM always assumes it is running in DAC mode. Anytime after this event, if the SLAVE mode SCM (currently running in DAC(slave) state) realizes that the remote SCM is not present, the SCM transitions to MASTER mode (now running in DAC(master) mode) and then performs a FAILOVER transition is performed to enter DEGRADED mode. If the MASTER mode SCM detects that the remote SCM is not operating, its can directly perform a FAILOVER transition to DEGRADED mode.

DEGRADED - [steady state] this indicates that the remote SCM is not operational requiring the local SCM to perform the duties of the faulted SCM. The local SCM is always running in MASTER mode while executing in a state of DEGRADED. The local SCM will remain in this state until the remote SCM begins responding to the local SCM's heartbeat messages. The local SCM will perform a transition to DAC mode only after determining that the remote SCM is operational. While in this state, the SCM continually attempts to contact a remote SCM

MONITOR - [fail state] - the SCM enters this mode if the configuration information is corrupted or does not make sense. It provides an opportunity for the operator to correct the situation. While in this state, the SCM is available for administration through the CML only. This state is entered from the INIT state. This transition, FAULT, shuts down all HASM operations, the transition,

name_OpMode(SCM_OPMODE_MASTER, SYSCONFIG_DEGRADED) is invoked.

FAILBACK - The SCM has detected that the remote SCM is now operational when this transition is performed. This transition will perform the necessary tasks to allow the local SCM to transfer control of the remote-owned resources back to the remote and to allow the remote-offered services to be re-continued by the remote SCM. On this transition, name_OpMode (SCM_OPMODE_MASTER, SYSCONFIG_DAC) is invoked.

IMPERSONATE - The SLAVE SCM impersonates the MASTER SCM if the MASTER SCM has faulted or the MASTER SCM impersonates the SLAVE SCM if the configured SLAVE mode SCM is running in DEGRADED mode. On SLAVE to MASTER transitions, name_OpMode(SCM_OPMODE_MASTER, SYSCONFIG_DAC) is invoked, On MASTER to SLAVE transitions, name_OpMode(SCM_OPMODE_SLAVE, SYSCONFIG_ASSUMEDAC) is invoked.

SHUTDOWN - The SCM transitions from an operational state to a halted state. The SCM will remain in the HALTED state until the SCM has been powered off and back on or if the reset button on the SCM is pressed. This transition can be executed on command through the SAM or through receiving notice that AC power has been lost and we are running on our alternate power source (UPC). On this transition, name_Stop() followed by name_ShutDown() is invoked.

RESET - The SCM transitions from an operational state to the BOOT state.

POWER OFF - The power switch is turned off or AC power is lost to the SCM.

FAULT - A transition to MONITOR mode is required. This transition invokes, name_Stop() invocations to halt operation of the SCM. Only the administration interface and any modules required of it will be left in an operating condition such that the SCM can be administered. The PSO will need to break from the cluster and remain in single SCM mode.

FIG. 4 depicts a state diagram 400 for the storage system 100. The SCM starts in a powered off state 402, OFF. It transitions to a powered on state 404 through a POWER ON transition 406. The SCM may enter the powered off state 402 at anytime through a POWER OFF transition 408. Once the SCM enters the powered on state 404, it jumps to the initialization state 410. Upon entering the initialization state 410, the SCM jumps to the BOOT state 412 and begins booting up the SCM. Upon completion of the BOOT, the SCM will transition to the INIT state 414 dependent upon which configured operational mode the SCM is configured for. Either the SCM transitions to the INIT(master) state 414 or the INIT(slave) state 416 depending on if it is configured to run in the MASTER mode or the SLAVE mode, respectively.

Once initialized, the master mode SCM transitions to a DAC master state 418 where it will operate until one of the SCMs fail. The slave mode SCM transitions from the initialization state 416 to the DAC slave state 420 where it will operate until one of the SCMs fail. If the master mode SCM fails, the slave mode SCM transitions along the IMPERSONATE transition 422 to enter a DAC master state 418. Then, because the master SCM must provide services to all the clients (i.e., the master processes requests for both

SCMs), the master SCM transitions to the DEGRADED state 424. If the slave mode SCM fails, the master mode SCM transitions to the DEGRADED state 424. When a failed SCM recovers, the master SCM transitions from the DEGRADED state 424 to the DAC master state. A recovered SCM always boots to the slave state 420.

The fail state 426 contains sub-states monitor 428 and halted 430. From steady state operation, a shutdown transition 432 causes the system to enter the halted state 430. From the fail state 426, if user chooses to restart the SCM or if the reset button is pressed, the SCM transitions along a reboot path 434 to the initialization state 404. If, in the initialization state 404, a fault occurs, the SCM transitions along path 436 to the fault state 426. In the monitor state 428, a minimal amount of functionality is enabled to allow system diagnostics to be performed on a failed SCM, i.e., read the event log, access certain files, and the like.

Normal Boot

FIG. 5 depicts a flow diagram 500 of the boot sequence for a normal system boot.

Initially, at steps 502 and 504 both SCMs are off and are powered on at the same time at steps 506 and 508. Both SCMs boot at steps 510 and 512. After booting, at steps 514 and 516, the SCMs examine their respective configuration files to see which operational mode they were configured to run in. The SCMs then, at steps 518 and 520, run in the appropriate INIT state. During initialization, NOP and GETSTATE messages are exchanged between the two SCMs in order to determine the existence, operability and operational mode of the remote SCM. At steps 522 and 524, READY transition is then performed which brings the SCMs into DAC mode. The storage system is now ready to accept

and complete requests from the network community, i.e., SCM A is now operating in master mode 526 and SCM B is operating in slave mode 528.

Booting with Faulted SLAVE SCM

FIG. 6 depicts a flow diagram 600 of the boot sequence with a SLAVE SCM that is faulted.

The operation SCM (SCM A) boot is normal until it reaches the INIT(master) state. The operational SCM discovers that the remote SCM (SCM B) is not responding. SCM A then decides after a period of time to continue with its boot process. Once the boot process reaches DAC(master) mode 526, the SM detects that no status messages are received or transmitted so it initiates a FAILOVER operation at step 602. The operational SCM transitions at step 604 to a DEGRADED state where it provides the services for both SCMs to the network community.

Booting with a faulted MASTER SCM

FIG. 7 depicts a flow diagram 700 of the boot sequence with a MASTER SCM (SCM A) that has faulted.

This sequence is similar to the previous scenario except that the configured MASTER SCM is faulted. The configured SLAVE SCM must perform at step 702 an IMPERSONATE transition to change from DAC(slave) to DAC(master) mode (step 704) before proceeding with the FAILOVER procedure at step 706. The operational SCM (SCM B) then operates at step 708 in the degraded state.

SLAVE SCM Faults

FIG. 8 depicts a flow diagram 800 of a process that occurs when the SLAVE SCM (SCM B) faults forcing the MASTER SCM (SCM A) to perform a failover operation.

If the SLAVE SCM faults (e.g., status messages cease to be generated) at step 806, the MASTER SCM simply performs a FAILOVER transition at step 802 to change the state of the MASTER SCM from DAC(master) to DEGRADED. At step 804, the MASTER SCM now provides services for both SCMs to the network community. Since the operational SCM has shared the failed SCMs configuration files and application state information, the failover transition is seamless.

MASTER SCM faults

FIG. 9 depicts a flow diagram 900 of a process that occurs when the MASTER SCM (SCM A) faults forcing the SLAVE SCM (SCM B) to impersonate the MASTER SCM and perform a failover operation.

This scenario is similar to the previous one except that the SLAVE SCM must transition to an operational mode of MASTER before performing the FAILOVER procedure. As such, when no status messages are detected because the master SCM has faulted at step 902, the slave SCM transitions at step 904 to the impersonate state. Then, at step 906, the operational SCM transitions into DAC master state. Finally, at step 908, the SCM transitions through a failover transition to a degraded state 910.

SLAVE SCM Resumes

FIG. 10 depicts a flow diagram 1000 of a process that occurs when the MASTER SCM (SCM A) is running in DEGRADED mode while the SLAVE SCM (SCM B) resumes, the MASTER SCM transitions to DAC(master) mode while the SLAVE SCM

transitions to DAC(slave) mode thus bringing the system back to DAC mode.

At step 804, the MASTER SCM (SCM A) is running in DEGRADED mode, while the SLAVE SCM (SCM B) is booting and initializing as described with respect to FIG. 5 above. When the slave SCM sends a ready signal to the master SCM during the INIT state, the master transitions at step 1002 from the degraded state 804 to the DAC master state 526. Once in the DAC master state and the resources that are used by the slave SCM have been released by the master SCM, the master SCM instructs the slave SCM to proceed. The slave SCM then transitions at step 524 to the DAC slave state 528.

MASTER SCM Resumes

FIG. 11 depicts a flow diagram 1100 of a process that occurs when the configured SLAVE SCM is running in DEGRADED mode while the configured MASTER SCM resumes operation. At step 1110, the operational SCM is operating in a degraded state. The failed SCM boots as described in FIG. 5 using steps 502, 506, 510, 514 and 518. Note that SCM A is configured to be a master SCM, so when booting, this SCM initializes as a master SCM. During the INIT state, SCM A requests the state of SCM B and is informed of its degraded state. SCM then transitions at step 1102 to impersonate a slave SCM and then at step 1104 enters the INIT(slave) state. During the INIT(slave) state, SCM A informs SCM B that SCM A is ready to operate. SCM B, at step 1112, begins a failback transition to provide SCM A with information it needs to carry on with operations previously handled by SCM B. Additionally, SCM B relinquishes control of certain resources for SCM A to use. SCM B then enters the DAC(master) state 1114. SCM B sends a proceed command

to SCM A and SCM A transitions, at step 1106, to the DAC(slave) state 1108.

Initialization

Upon initialization, the HASC needs to establish the operational mode of the SCM (MASTER or SLAVE), the operational mode of the storage system (DAC or DEGRADED), and the integrity of the configuration information. The operational mode of the SCM is first assumed to be the configured operational mode. This assumption is made until otherwise changed, i.e., the SCM will run in this assumed operational mode until the SCM establishes communications with the remote SCM and the operational state of the remote has been established. If the remote SCM is in degraded mode, the local SCM must run in SLAVE mode regardless of its configured operational mode. The HASC will invoke the IMPERSONATE transition function to transition the SCM into SLAVE mode if it is configured to operate in MASTER mode. If the remote SCM is booting, the local SCM uses its configured operational mode unless it conflicts with the remotes. In this case, a configuration error has occurred on the part of the system administrator. The SCMs will enter MONITOR mode to allow the system administrator to correct the problem.

The operational mode of the storage system is determined solely on the condition of the remote SCM. If the remote SCM is alive and operating correctly, a mode of DAC is established. If the remote SCM cannot be contacted or it is not responding correctly, a mode of DEGRADED is established.

The configuration information is established by examining the information from the CTCM regarding the state of the SCM configuration. The CTCM indicates if the

configuration information is properly synchronized and if it is consistent with the information stored on the logical storage devices. It will determine if the current SCM contains the latest configuration information and if not, will allow the SCM to retrieve the latest configuration information from the replicated configuration database after which a reboot operation will occur. The replicated configuration database is disclosed in detail in U.S. patent application serial number _____ filed simultaneously herewith (Attorney docket ECCS 008), which is incorporated herein by reference.

The HASC will invoke name_OpMode() function in the appropriate software modules after the operational mode of the SCM and the operational mode of the storage system has been established.

Committing Suicide

If the LHM has determined that the local SCM is no longer capable of operating correctly in its current state, the LHM will request that the HASC commit one of several types of suicide. If the LHM was determined that it is a software condition caused by a design or implementation fault, the HASC will simply reboot the SCM. The remote SCM will takeover control of the local SCM's resources and services. If the LHM has determined that a more permanent condition has arisen such as a hardware failure, the local SCM will be disabled until a power cycle is applied and the remote SCM will takeover the resources and services of the local SCM as well.

Status Monitor

The SM is responsible for monitoring the status messages of the remote SCM to determine if the remote SCM

is alive and operating properly. If the SM determines that the remote SCM is not operating correctly, it will notify the HASC to initiate a failover operation. The SM employs redundant channels in order to transmit and receive status messages.

FIG. 12 depicts a block diagram of an illustrative embodiment of a status monitor 1200. Specifically, he SM is divided into a client 1202 and server 1204 task. Each SCM employs both a client and a server. The client 1202 comprises a status message generator 1206, a TCP/IP stack 1208, a plurality of NIC drivers 1210 and a plurality of NICs 1212. The status message 1202 client is responsible for issuing status messages on a periodic basis. The messages are coupled through a plurality of sockets 1214 to be broadcast on a plurality of network paths 1216. This client task status issues these messages once every second across all available network channels to the server 1204 in the remote SCM. This allows a verification of all network channels to ensure that both SCMs are connected to all networks. This is important because, if a SCM failure occurs, the remaining SCM must have access to all resources connected to the failed SCM. The client 1202 also updates the status information which contains the status of all the network channels.

The server 1204 comprises a status message receiver 1218, a status API 1220, a status analyzer 1222, a fault analyzer 1224, a status information database 1226, and a network communications portion 1228. The network communications portion 1228 comprises a plurality of sockets 1230, a TCP/IP stack 1232, a plurality of NIC drivers 1234 and NICs 1234. The server 1204 is an iterative server and listens for status messages on the set of sockets 1230 to all the available network interfaces and performs analysis on the state of the various network channels over which status messages are received. The

server 1204 updates the status information database 1226 every time that a status message is received from the client 1202 running on the remote SCM. The status information database 1226 contains the current state of each network port. The status analyzer 1222 checks the status information database 1226 on a periodical basis. The status analyzer 1222 is looking for network ports that are not being updated. An un-updated network channel status indicates that some sort of fault has occurred. The status analyzer 1222 calls the fault analyzer 1224 to analyze the situation. The fault analyzer 1224 is also responsible for updating the network port objects through a socket 1238 coupled to the TCP/IP stack 1232 and the remote SCM configuration object. The status API 1220 allows the status of the status monitor 1220 to be returned. Information regarding the status monitor 1200 as well as the network channel state and remote SCM state are available.

If no status messages are being received from the remote SCM, the SCM assumes that the remote SCM has failed. The HASC is notified of this condition.

If one of the host network ports is not working properly, status messages issued over the inoperative channel are not received by the status server message. An event is logged to an event notification service. If the dedicated SCM channel is not operational, no actions are taken other than the notification of the event. If one of the Host network connections has become inoperative, the status monitor 1200 in conjunction with the remote SCM's status monitor attempt to determine the location of the fault as the local SCM's network port, the cabling between the local SCM and the network, the network is down (hub has failed), the remote SCM's network port has failed, or the remote SCM's network cable has failed.

The status monitor communicates through a special RSCM interface designed for the status monitor. This special interface allows better control over the communications channel so that the status monitor can better perform its job function.

The server will need to wait on several sockets using the SELECT command. Every time a status message is received, the sequence number is stored and the count information is incremented by the difference between the current sequence number and the last sequence number. The time-out value is 1 second. Every second, the status analyzer function is run to adjust the status information. It decrements the network channel count information by 1. If the count information hits 0, this indicates the network channel is not working. The starting value for the network channel count information will start at 10. It may need to be adjusted later.

The API allows another task to inquire about the status of the network connections and the remote SCM. The API returns a GOOD/BAD indication of each network connection as well as for the remote SCM. Statistically information must also be returned regarding number of packets send/received, number of missing packets and on which network connections.

One embodiment of a status monitor is described in U.S. patent application serial number _____ filed simultaneously herewith, (Attorney docket ECCS 006), which is incorporated herein by reference. The present invention may utilize the foregoing status monitor technique or any other technique that facilitates identification of a faulted remote SCM.

Local Health Monitor

The local health monitor is part of the HASM that monitors and assesses the health of the local SCM. It basically looks at the current state of operation and ascertains the health of the SCM. If the LHM determines that the SCM is not running in a sane state, its logs the condition to the event notification service and then notifies the HASC that it recommends that the SCM surrender its resources and services to the remote SCM and to initiate a reboot operation (suicide module). The reboot operation is intended to reinitialize the SCM in hopes that the problem could be corrected through this re-initialization process. If an application task is running and it recognizes an unrecoverable error condition, it has the ability to call the LHM. Additionally, if a fault condition occurs such as a divide by zero error, the LHM will intercept the fault and forward the request to the HASC which will reboot the system.

The LHM also gathers environment status information from the EMM and uses this in its determination of the health of the local SCM. If the EMM has determined that the system is in danger of overheating, it can initiate a shutdown operation to prevent data from being corrupted.

The LHM is also able to monitor all the services tasks that are running and have the ability to restart them in case the server task terminates due to an unrecoverable error.

Applications that upon detection of an unrecoverable situation that requires re-initializing the SCM, the PANIC function may be called.

On a periodic basis, the LHM generates a wellness message which is transmitted through the event notification service. This wellness message is intended to convey that

the storage system is working normally. The system administration module supplies an API that administers the interval upon which this message is transmitted.

SCM Software Architecture

FIG. 13 diagrammatically depicts the relationship of the main system software components. The RSCM 308 sits on top of the TCP/IP stack 1302 which it uses as a transport for communicating with a remote SCM. The RSCM 308 consists of two layers, the remote SCM communications API 1308 and the redundant link management 1310.

The Remote SCM Communications Manager (RSCM) is responsible for maintaining a reliable communications link with the remote SCM. This module provides a service which allows other software modules and applications in the software architecture to communicate with the remote SCM. This module also provides a redundant link management layer which is responsible for managing the status information regarding the various channels used for communications between the SCMs. The RSCM provides a variety of communications mechanisms. It provides synchronization primitives and synchronous and asynchronous message passing. The RSCM module provides the capability of using physical interfaces such as serial (RS232-C), disk block communications, fibre channel, memory mapped (VIA), and more.

The API 1308 provides an abstraction layer to the addressing problems in working with redundant communications links between a client and a server. This API 1308 simplifies the process of establishing a communications channel, reading and writing data, and terminating a connection to a peer application running on the remote SCM. All common network programming APIs (client/server applications 1312 and status monitor

applications 1314) are encapsulated within this layer. This allows errors to be recorded internally and decisions to be made regarding the choice of network channels without intervention or knowledge of the invoking task. This status information is passed on down to the redundant link manager 1310 that records and collates this information.

The redundant link management (RLM) is responsible for determining the configuration of networks between the SCMs and providing a decision function that decides which network port to use when a channel is opened to a remote server. The RLM updates information related to the configuration of the network ports, their status, and error statistics. This information is stored in the RLM module.

The RLM is responsible for recording error statistics regarding the various network channels and providing a selection criteria for selecting a communications channel for a network application. The RLM is coupled via a socket 1316 to a file system layer 1305 (stack OS) (the TCP/IP stack 1302 and the NIC drivers 1304). Underlying the file system layer 1305 is an operating system (VxWorks) 1306. Anytime an error occurs on a socket connection to the remote SCM, the RLM looks up the sockets IP address and use it to determine which network channel is having the problem. The RLM then updates the NIC statistics for the appropriate NIC. Anytime that a client application needs to open a socket to the remote SCM, this module will determine which network channel is the best one to use. The statistics for each network port are stored in the appropriate NIC object which is managed by the RLM module. These statistics are updated by both the RSCM in case of an error and by the SM. The RSCM records only operational errors whereas the SM records both operational and time domain errors (status message not received in time).

Persistent Shared Objects (PSO) are used as a paradigm for sharing data between the SCMs. A PSO is an object

whose value is persistent across power cycles and that is accessible from any SCM in the storage system cluster. The PSO manager 310 is responsible for maintaining the coherency of the information between the various SCMs. Persistence is implemented by storing the PSO in the root file system of each SCM. The object name is identical regardless of which SCM accesses it.

The PSO manager 310 will sit on top of the remote SCM communications module which it uses for all communications between itself and the remote SCM.

A shared object is referenced by its name or its ObjectID. The name is an ASCII string that contains the name of the object in english. The ObjectID is an opaque value assigned by the PSO Manager and is used to reference the shared object for all operations. The ObjectID is actually a pointer to the object. The ObjectID allows the object to be quickly referenced. The ObjectID is not guaranteed to be identical from power cycle to power cycle. It is also not guaranteed to be the same from SCM to SCM.

A shared object must be created before it can be used. The creation process allocates the resources required by the object and return an ObjectID to the creation task. If a shared object is already created, the calling task should call the pso_Exist() function first to see if the object exists already. A shared object has an attribute called persistence. If a shared object is persistent, this indicates that the shared object maintains its existence through storage system power cycles. If a shared object is not persistent, it must be re-created every time the storage system is powered up.

Before a shared object is to be written, it must be locked for exclusive access. This lock extends through the set of SCMs participating in the shared object global name space. The lock guarantees that the lock owner has the exclusive right to modify the shared object. Upon

completion of the modifications to the shared object, the task must unlock the object to allow other local or remote tasks to gain the right to modify it.

A copy of the shared object will exist on all SCMs. Anytime that value of a shared object is changed, the changed object must be distributed to all the other SCMs participating in this distributed application. The PSO Manager maintains a database of the currently operating SCMs. This database is used as the distribution list when a shared object needs to be updated. All SCMs must be aware of all other participating SCMs.

A shared object must possess a locking mechanism that prevents multiple writers from updating the object incorrectly. This locking mechanism will guarantee mutual exclusion and must be able to lock across SCM extents. A binary semaphore must be associated with each copy of a shared object.

The shared file module 312 is responsible for synchronizing the information stored in two different files on one or two systems. Whenever a file is written by the MASTER SCM, it is updated by calling the appropriate APIs in the SFM 312.

The configuration transaction control module (CTCM) 316 is responsible for maintaining the integrity of the configuration of the storage system. Its main responsibility is to maintain the shared replicated configuration database stored on the two SCMs and the storage arrays. The CTCM 316 is not responsible for maintaining the actual configuration files, but acts as a transaction wrapper around which configuration changes take place. Only the SCM running in MASTER mode is allowed to perform a configuration transaction. The SLAVE mode SCM must run under the direct supervision of the MASTER SCM. Only the MASTER can dictate to the SLAVE as to when its own timestamp files can be updates.

The CTCM 316 is invoked anytime one of the SCM configuration files is updated. Most of these calls originate from the SAM.

The configuration information is stored both on the SCM's internal hard drive (referred to as the local configuration database) in its usable format as well as on the private extent of a selected group of logical storage devices (referred to as the shared replicated configuration database). The shared replicated configuration database is considered the primary source of configuration information. It is the responsibility of the CTCM 316 to maintain consistency of information in the local and shared replicated configuration database such that the latest version of the configuration information can always be reliably extracted.

The CTCM 316 is able to ascertain on powerup, if the local configuration information is correct or not. If it is not the latest version or if the configuration information is corrupt, the CTCM can retrieve a copy of the latest configuration from the shared replicated configuration database and correct the corruption. This is implemented by performing a restore operation of a valid archive found on the storage array.

Although various embodiments which incorporate the teachings of the present invention have been shown and described in detail herein, those skilled in the art can readily devise many other varied embodiments that still incorporate these teachings.

What is claimed is:

1. A network appliance for providing network services to a plurality of clients comprising:

first means for communicating with said clients;

second means for communicating with network service equipment;

means, coupled to said first and second communicating means, for storing state and configuration information regarding a remote network appliance; and

means, coupled to said storing means, for utilizing said state and configuration information regarding said remote network appliance to cause said apparatus to perform services that are provided by said remote network appliance.

2. The network appliance of claim 1 wherein said first means for communicating comprises a network interface card and a network interface card driver.

3. The network appliance of claim 1 wherein said second means for communicating comprises SCSI channel equipment.

4. The network appliance of claim 1 wherein said second means for communicating comprises fiber channel equipment.

5. The network appliance of claim 1 further comprising a storage pool for storing data that can be accessed by the clients.

6. The network appliance of claim 1 further comprising means for monitoring a status of said remote network appliance.

7. The network appliance of claim 6 further comprising means for impersonating said remote appliance when said status indicates said remote network appliance is not operating properly.

8. The network appliance of claim 1 further comprising a means for monitoring a status of said network appliance and if said network appliance is not operating properly, re-booting said network appliance.

9. A storage system comprising:

a first storage controller module couple to a network;

a second storage controller module coupled to said network;

a storage pool coupled to said first and second storage controller modules;

where said first storage controller module stores configuration and state information about said second storage controller module and said second storage controller module stores configuration and status information about said first storage controller module.

10. The storage system of claim 9 wherein said first storage controller module comprises a status monitor that monitors the status of the second storage controller module, and said second storage controller module comprises a status monitor that monitors the status of the first storage controller module.

11. The storage system of claim 9 wherein said storage pool comprises a first storage array coupled to said first and second storage controller modules, and a second storage array coupled to said first and second storage controller modules.

12. The storage system of claim 11 wherein said storage arrays are coupled to said storage controller modules using either SCSI connections or a fiber channel network.

13. A method of operating a storage system having a first and second storage controller modules coupled to a storage pool, said method comprising:

executing an initialization state to boot a first storage controller module into a master state; and

executing an initialization state to boot a second storage controller module into a slave state.

14. The method of claim 13 further comprising:

operating said first and second storage controllers in a steady state until a fault is detected in one of said first or second storage controller modules;

causing an operational storage controller module to enter a degraded mode, wherein said operational storage controller module performs functions that are otherwise performed by said faulted storage controller module;

rebooting said faulted storage controller module.

15. The method of claim 14 further comprising:

causing said faulted storage controller module to reboot in a slave mode;

releasing resources from said operational storage controller module to enable said rebooted storage controller module to control said resources.

16. The method of claim 15 wherein said first and second storage controller modules share status and configuration information.

17. The method of claim 15 wherein said first and second storage controller modules analyze status information from a remote storage controller module.

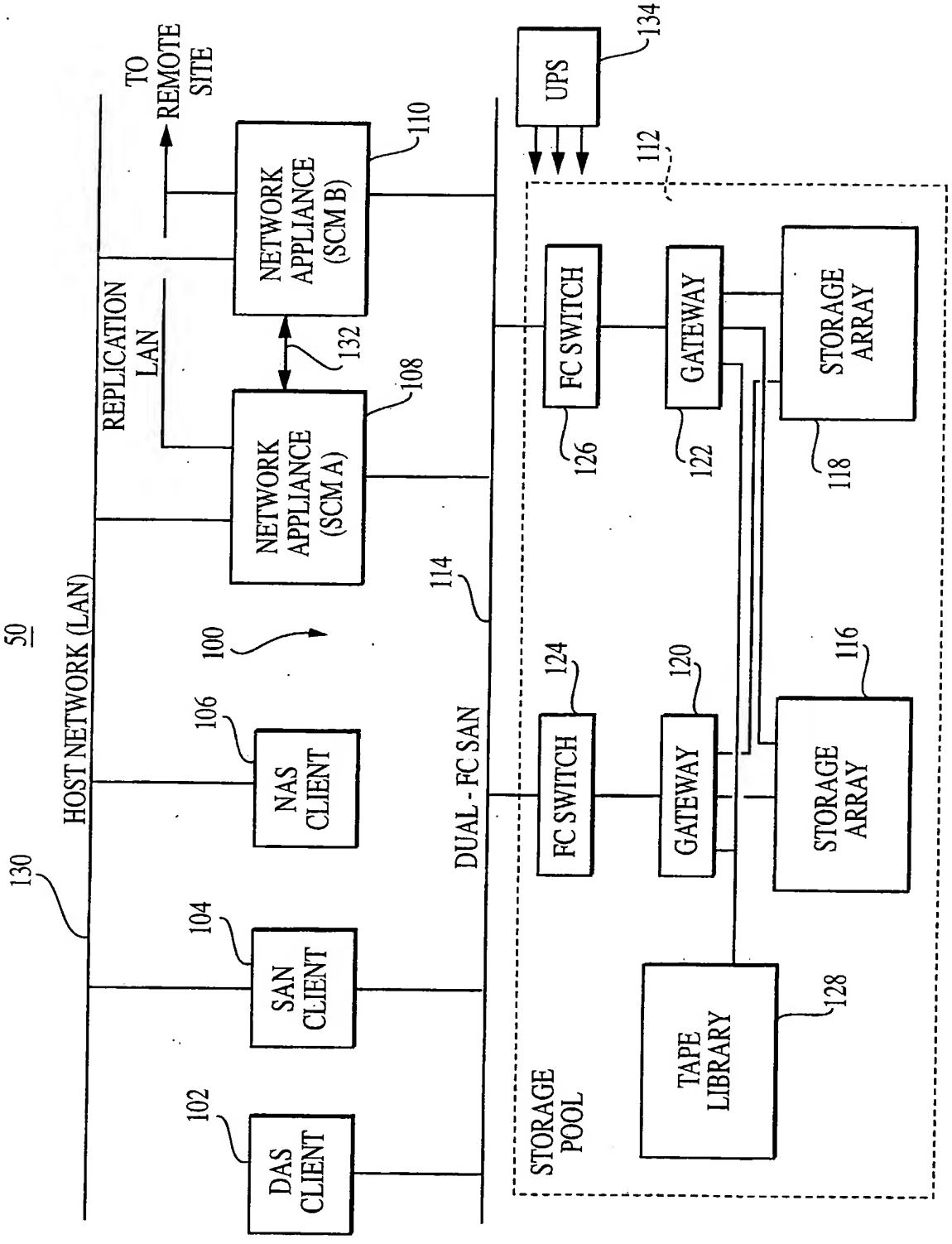


FIG. 1

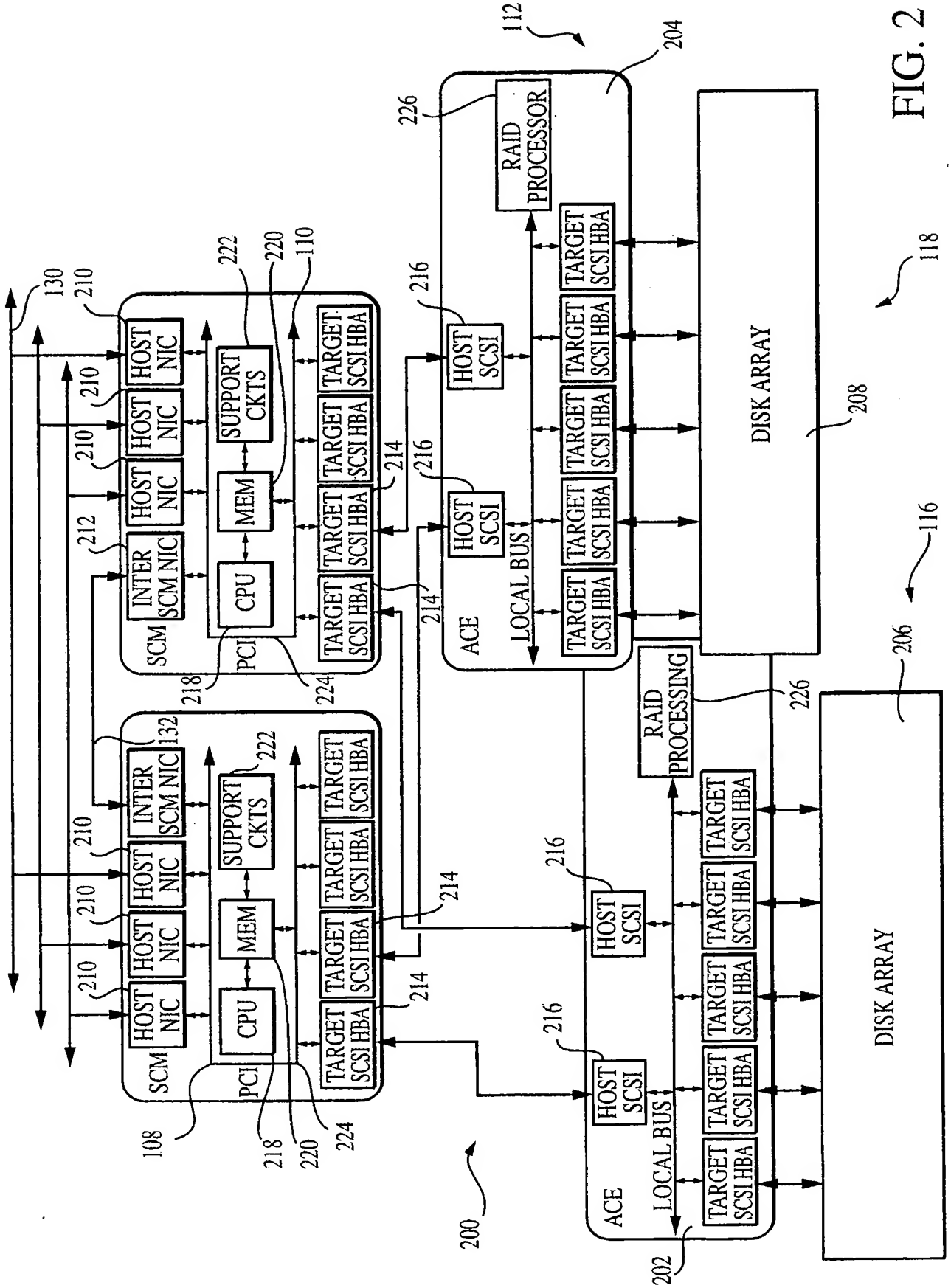

200
202
204
206
208
210
212
214
216
218
220
222
224
226
108
110
112
116
118
130
132
SCM
HOST NIC
INTER SCM NIC
CPU
MEM
SUPPORT CKTS
PCI
TARGET SCSI HBA
ACE
HOST SCSI
LOCAL BUS
RAID PROCESSOR
RAID PROCESSING
DISK ARRAY

FIG. 2

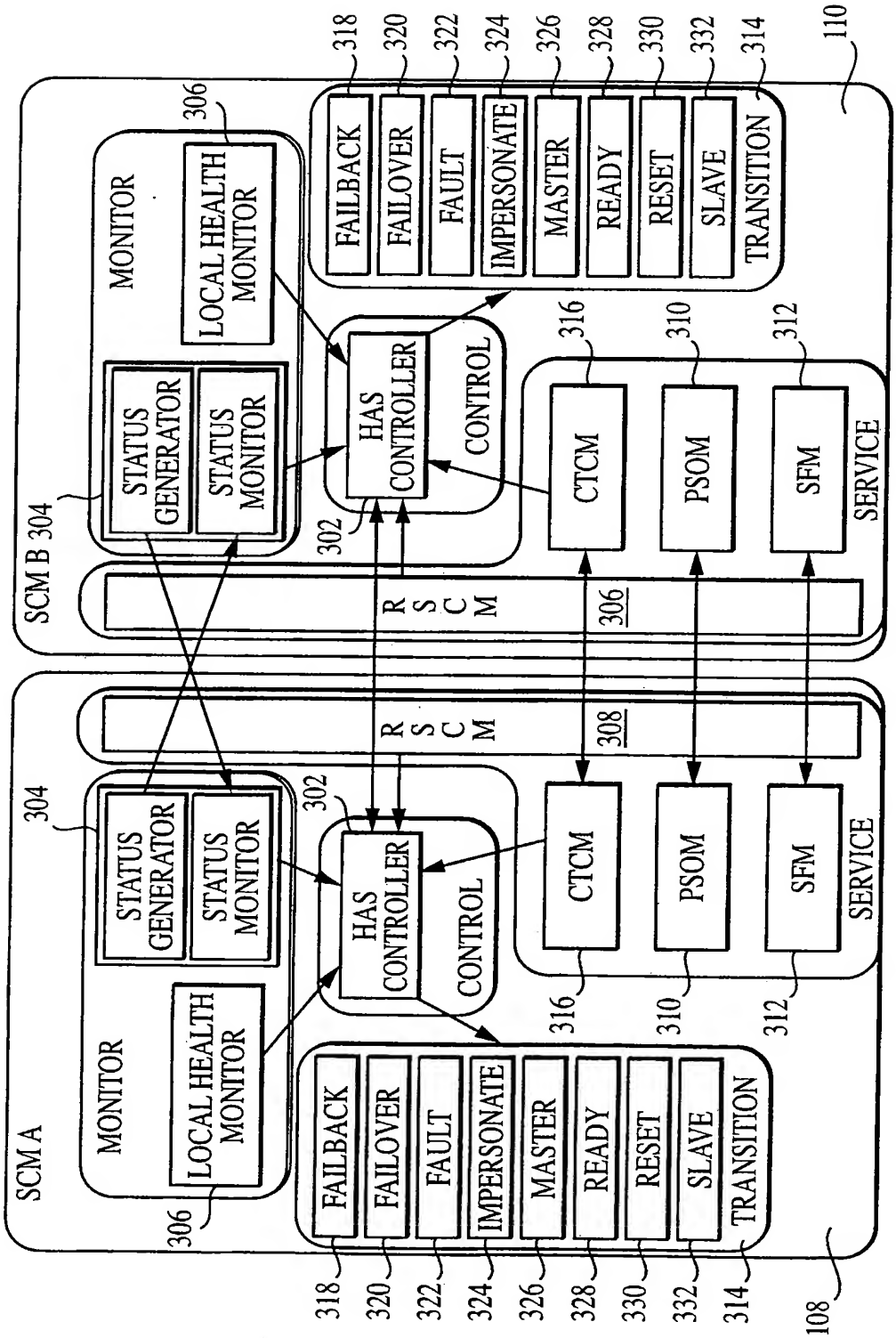


FIG. 3

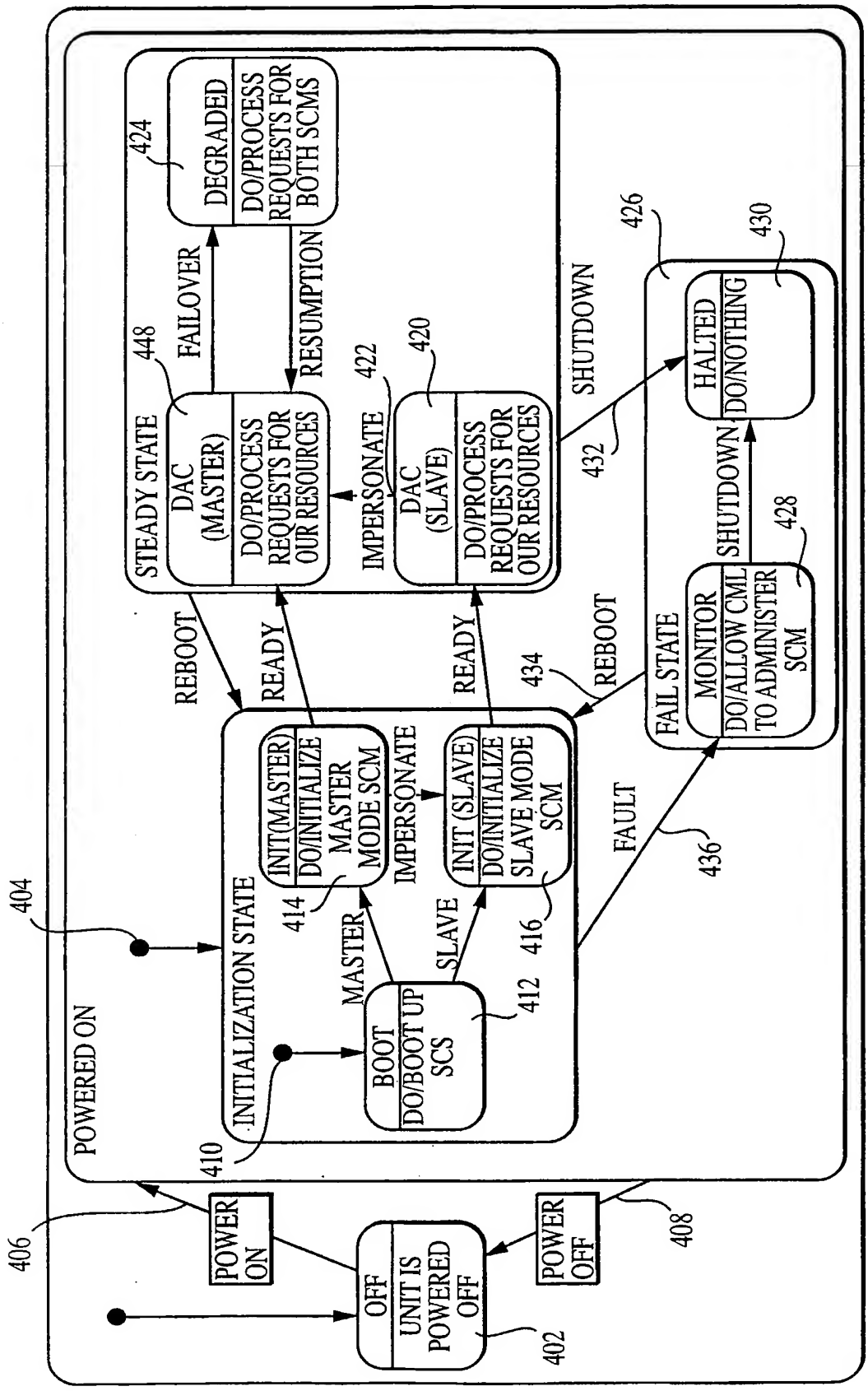


FIG. 4

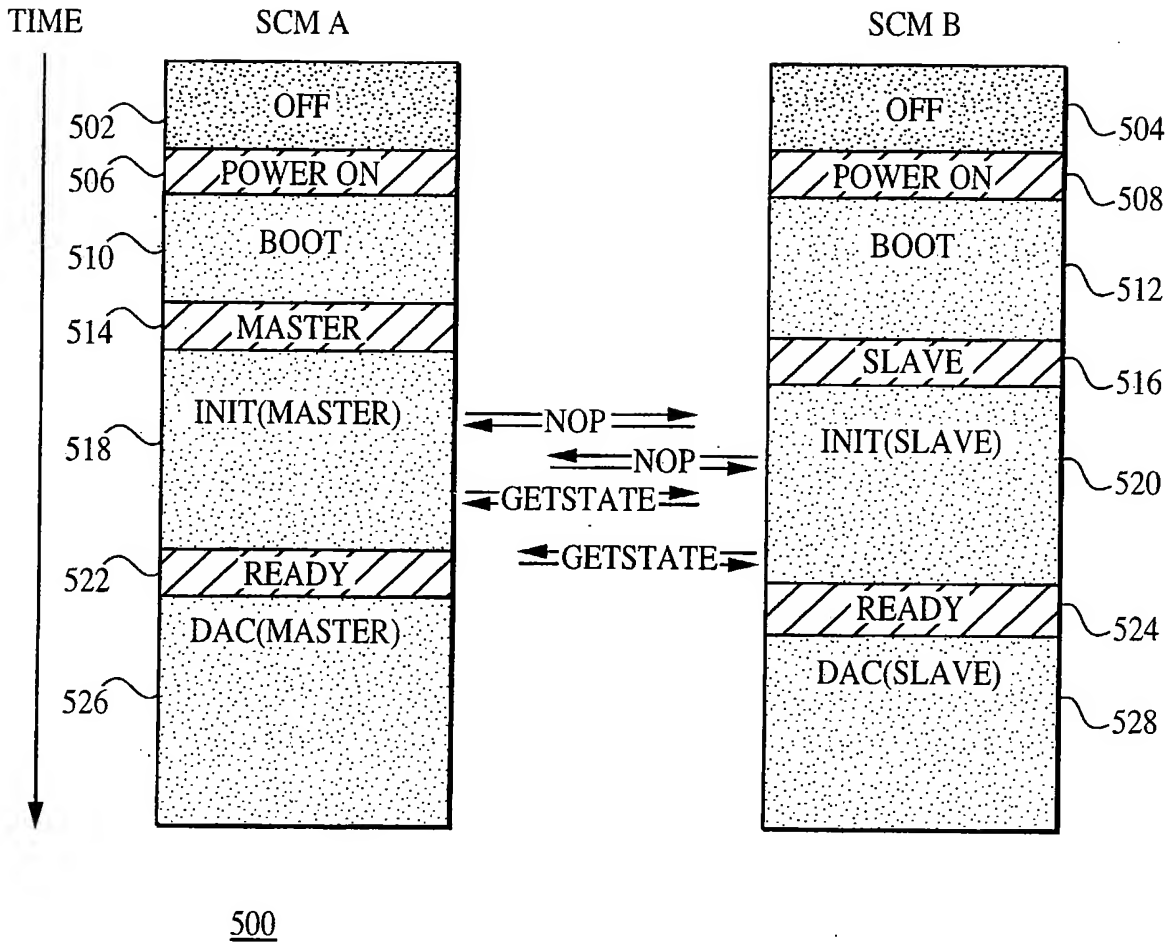


FIG. 5

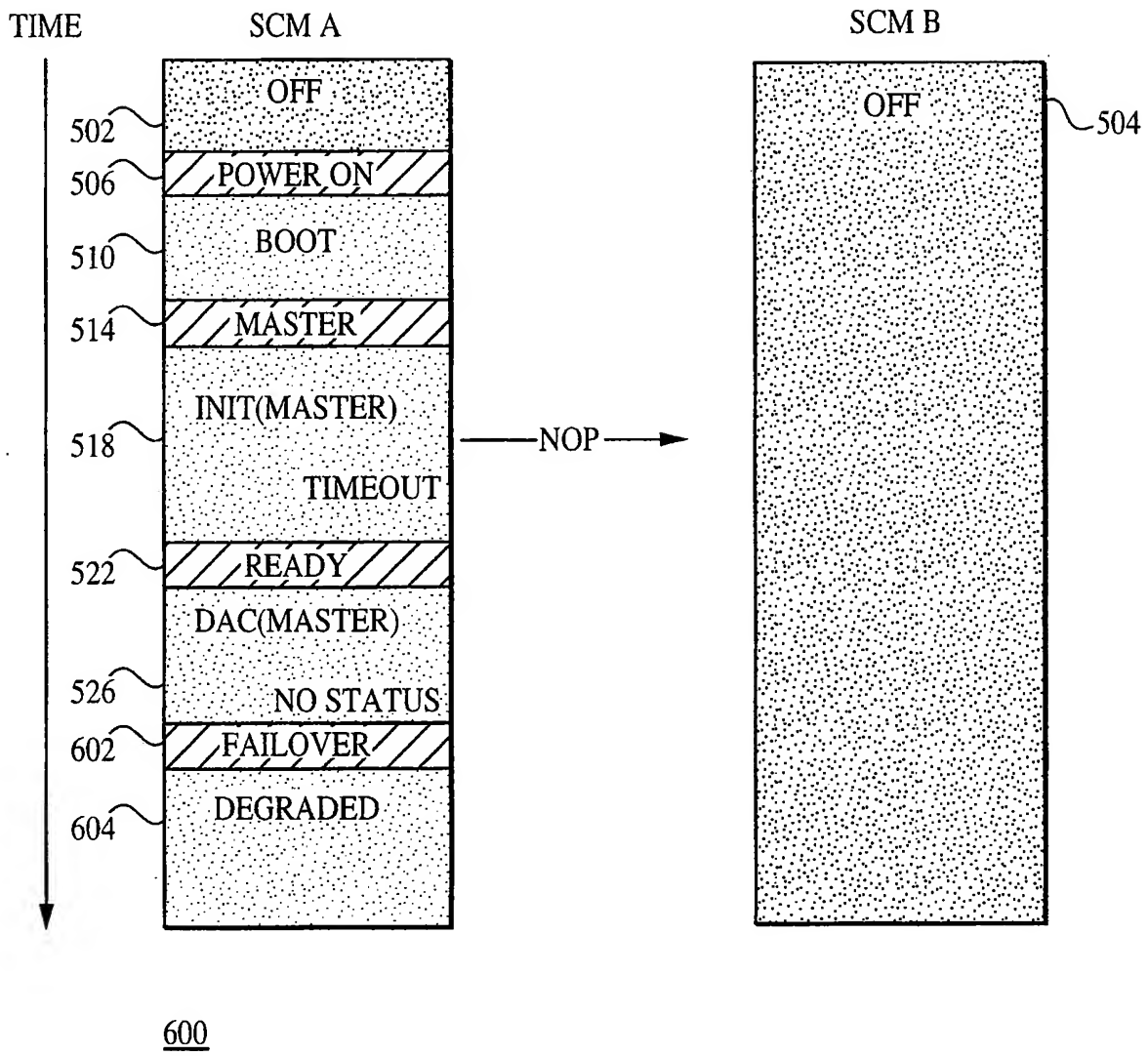

TIME
SCM A
SCM B
OFF
502
POWER ON
506
BOOT
510
MASTER
514
INIT(MASTER)
518
TIMEOUT
NOP
READY
522
DAC(MASTER)
NO STATUS
526
FAILOVER
602
DEGRADED
604
OFF
504
600

FIG. 6

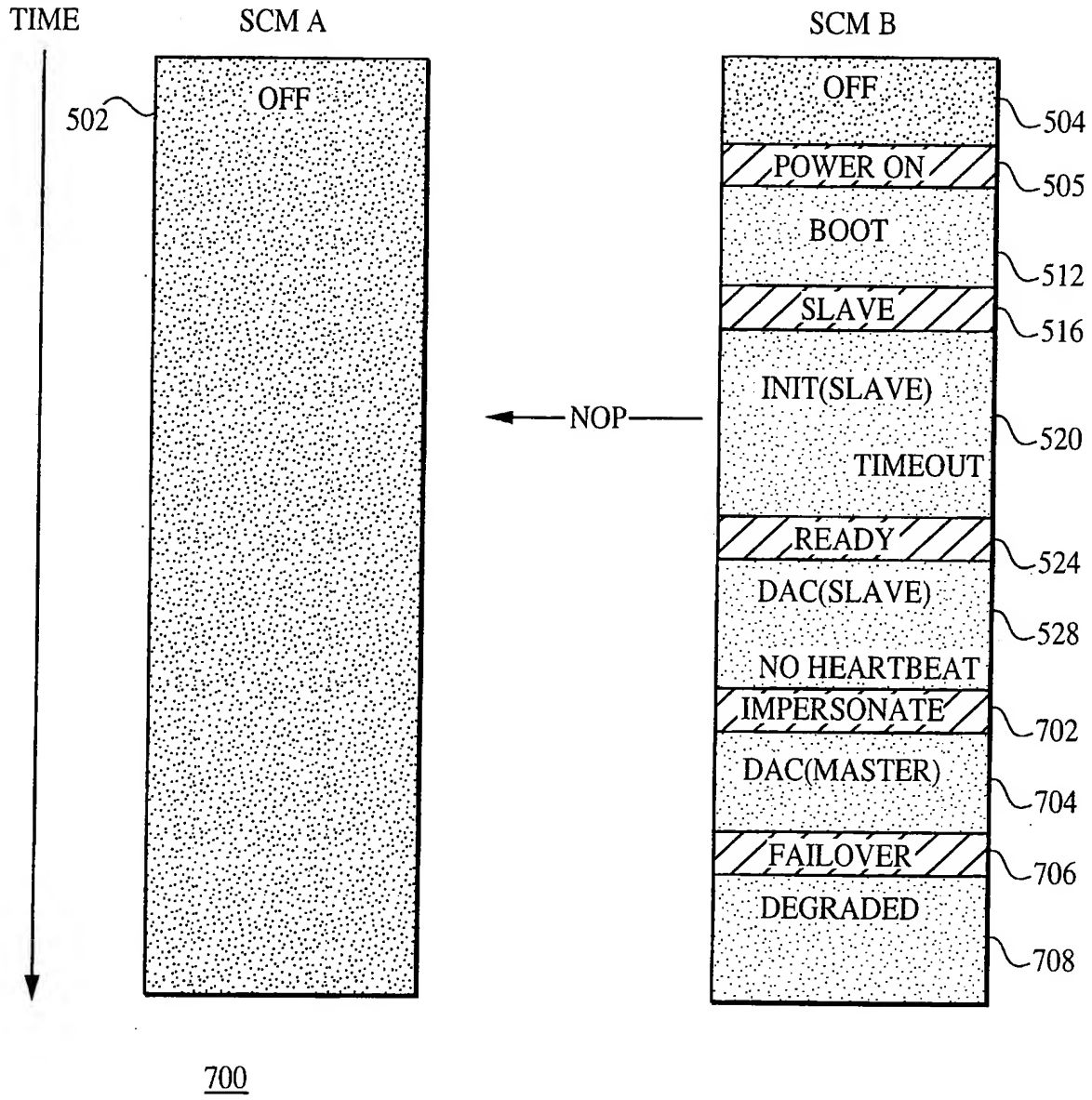


FIG. 7

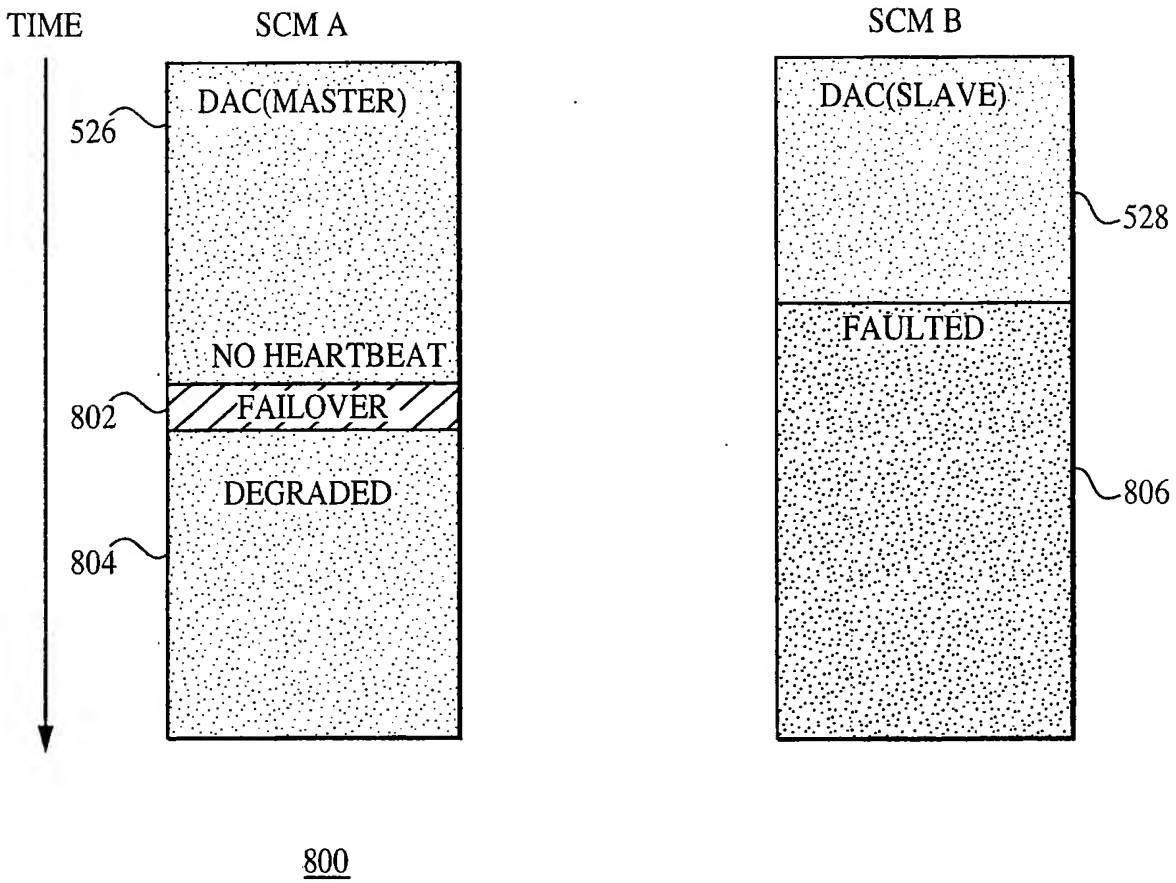

TIME
SCM A
SCM B
DAC(MASTER)
526
NO HEARTBEAT
FAILOVER
802
DEGRADED
804
DAC(SLAVE)
528
FAULTED
806
800

FIG. 8

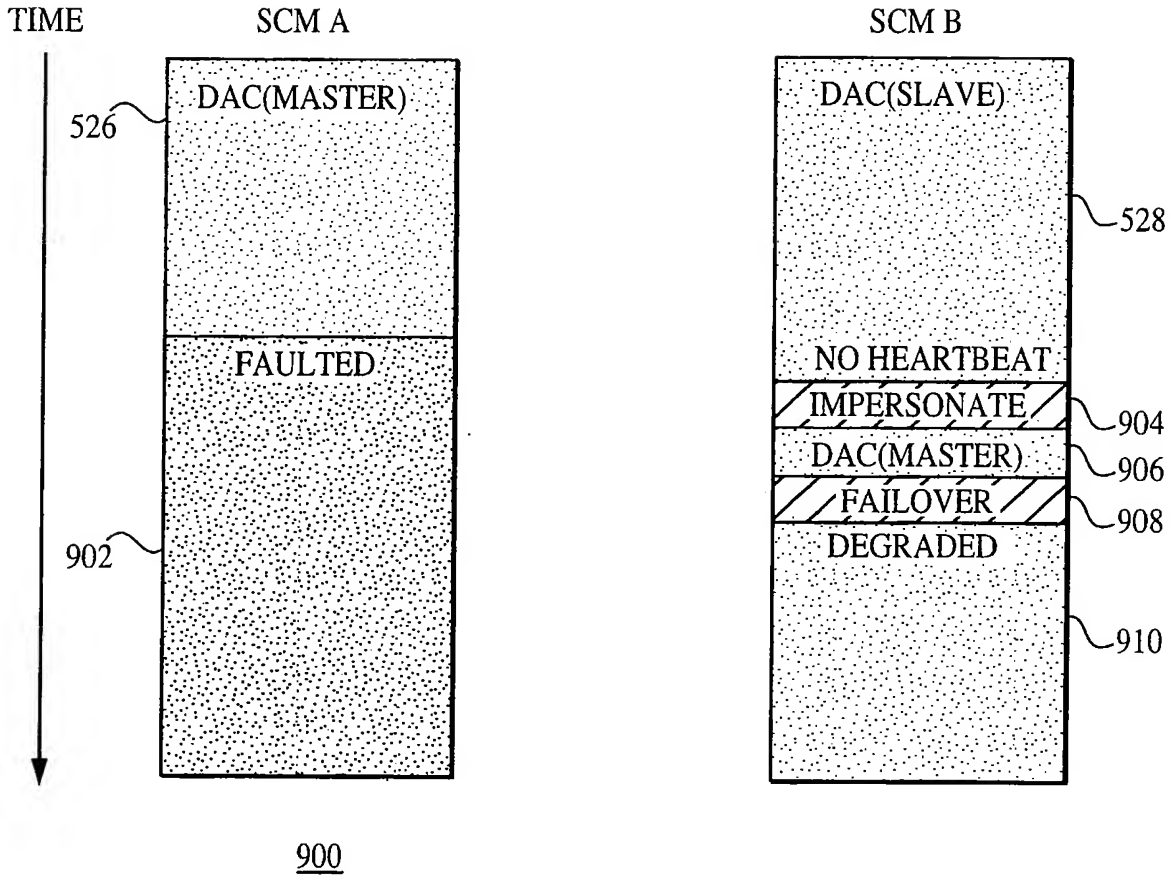


FIG. 9

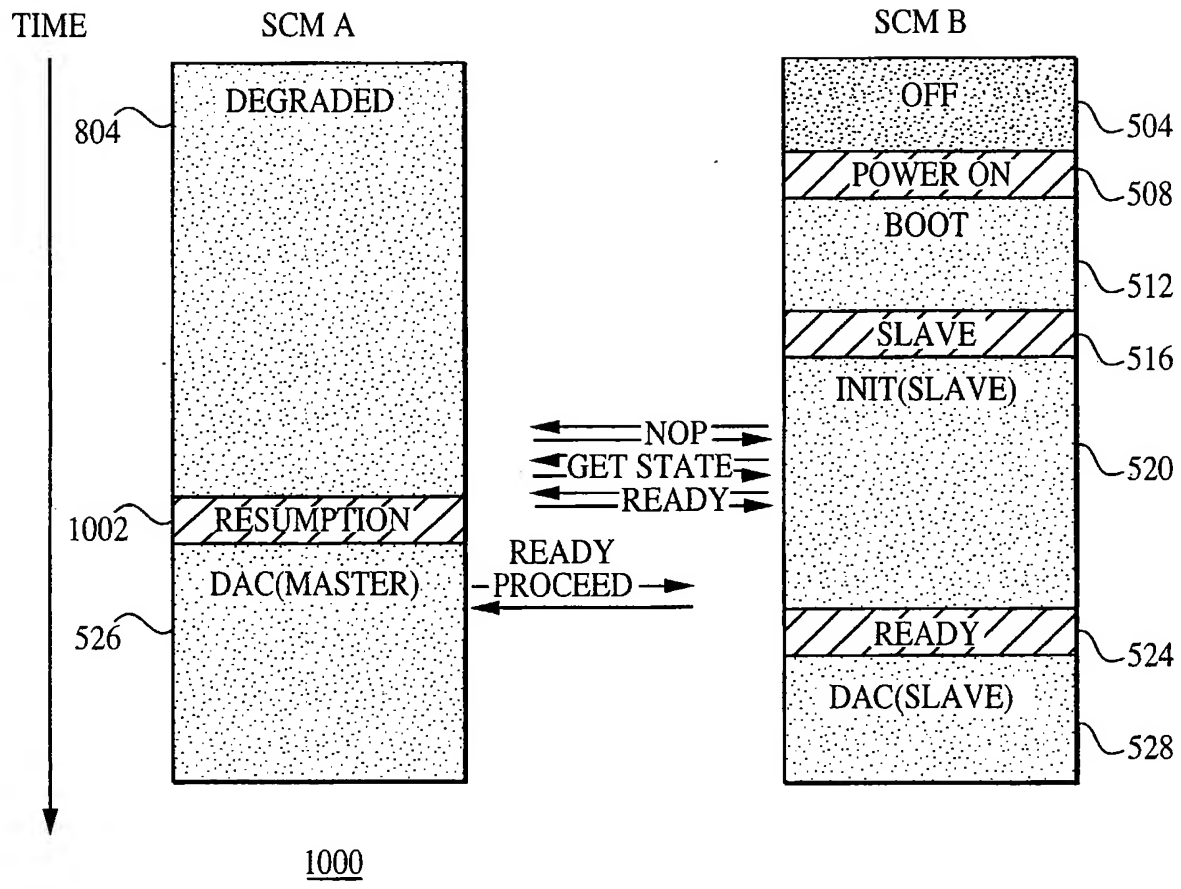
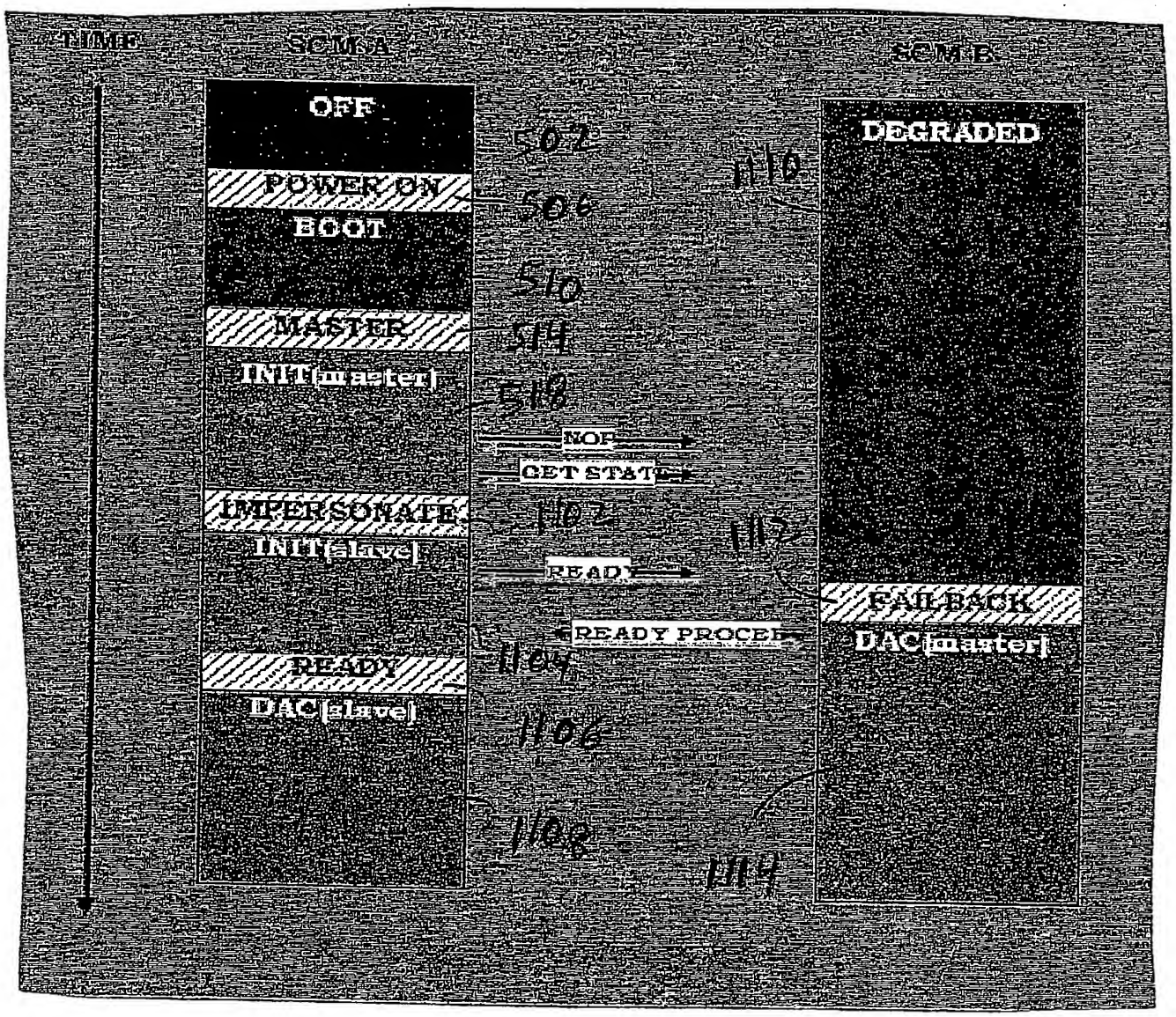

TIME
SCM A
SCM B
DEGRADED
804
OFF
504
POWER ON
508
BOOT
512
SLAVE
516
INIT(SLAVE)
NOP
GET STATE
READY
520
1002
RESUMPTION
READY
PROCEED
DAC(MASTER)
526
READY
524
DAC(SLAVE)
528
1000

FIG. 10



1100

FIG 11

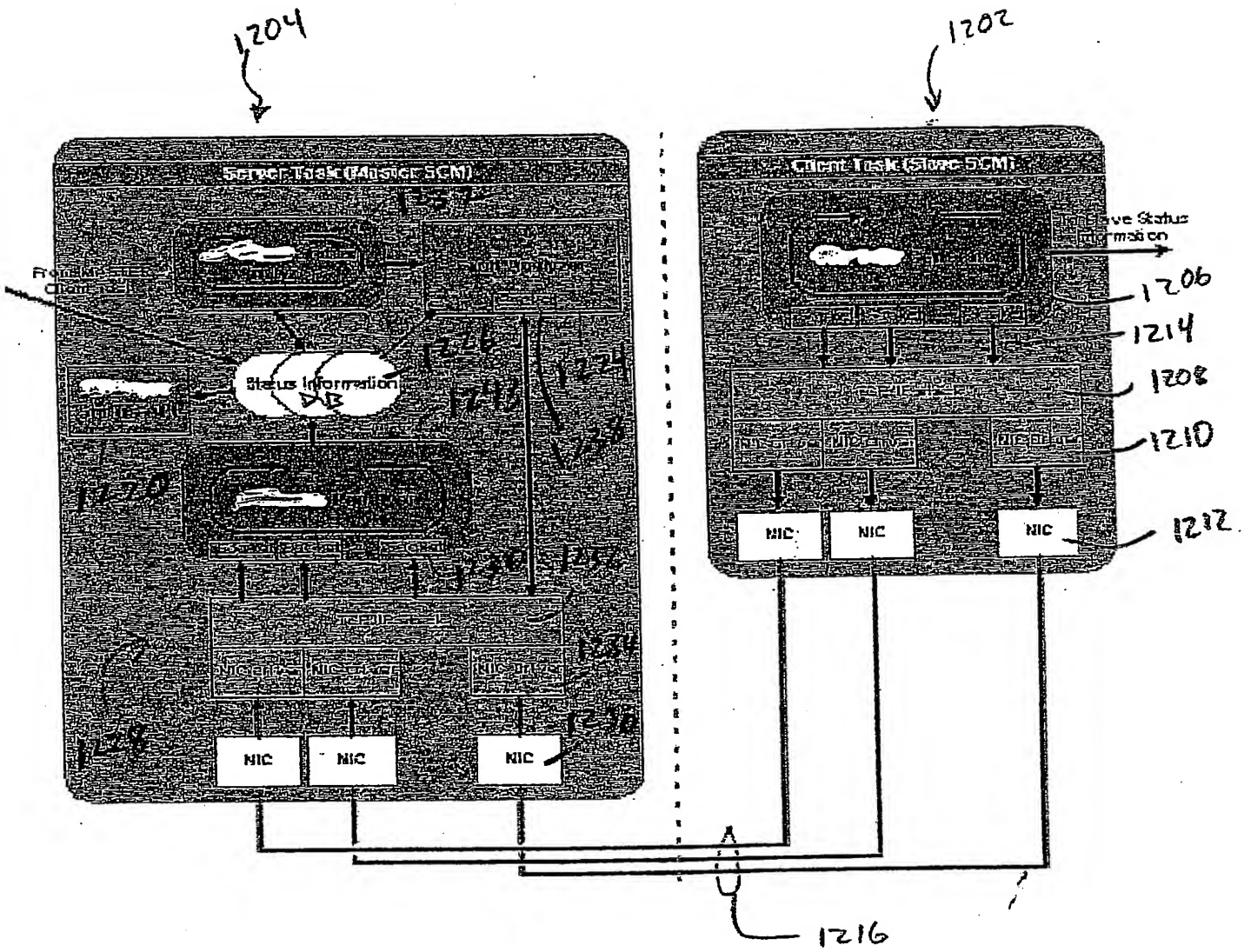

1204
1202
Server Task (Master SCM)
Client Task (Slave SCM)
Status Information
1206
1214
1208
1210
1212
1216
NIC
NIC
NIC
NIC
NIC
NIC
1200

FIG. 12

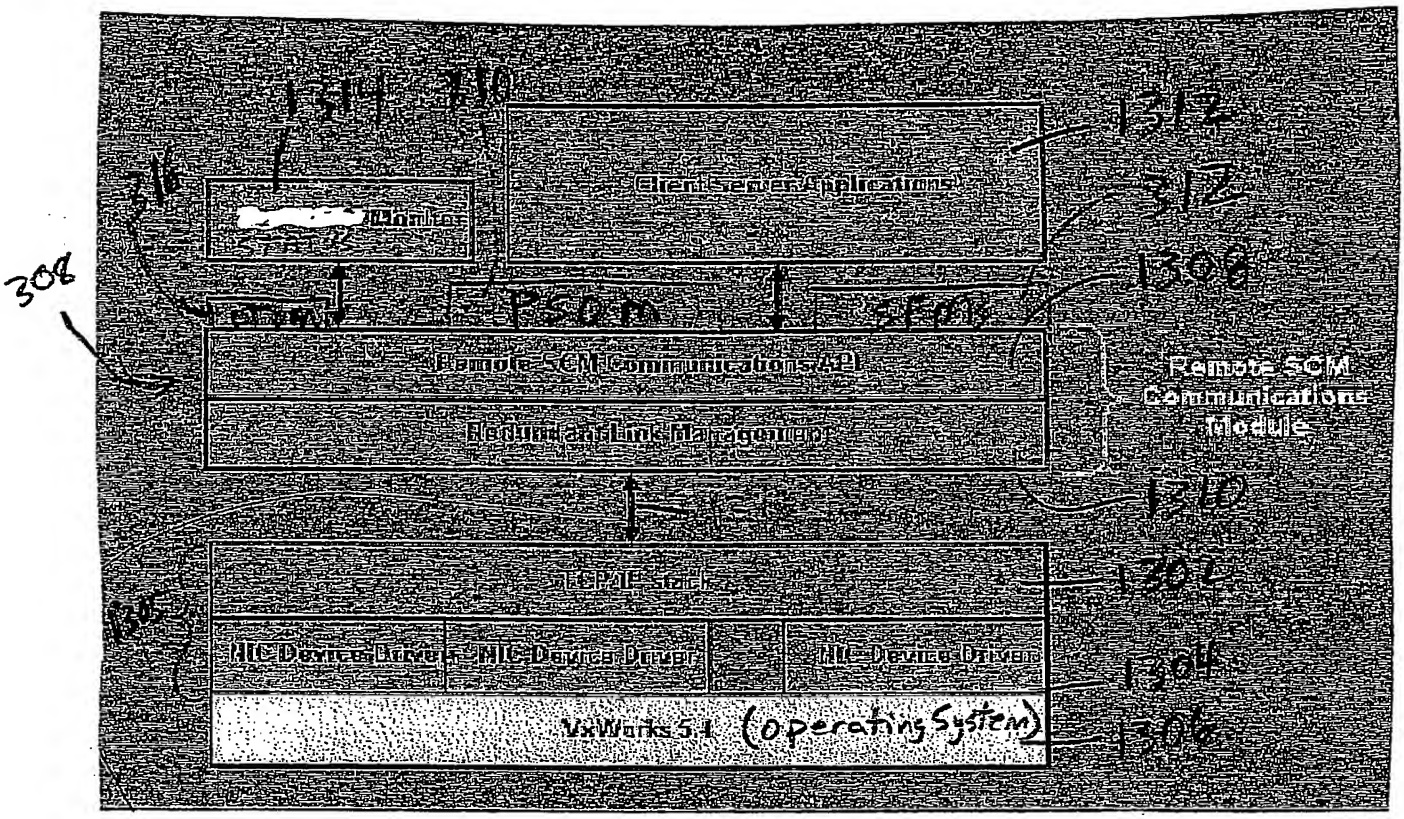


FIG. 13

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 November 2001 (01.11.2001)

PCT

(10) International Publication Number
WO 01/082080 A3

(51) International Patent Classification[7]: H04L 29/06, G06F 11/20

(21) International Application Number: PCT/US01/12889

(22) International Filing Date: 20 April 2001 (20.04.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/552,781 20 April 2000 (20.04.2000) US

(71) Applicant: CIPRICO, INC. [US/US]; Suite 60, 2800 Campus Drive, Plymouth, MN 55441 (US).

(72) Inventors: MCMILLAN, Ben, H., Jr.; 125 Marvin Road, Middletown, NJ 07748 (US). DAVIS, Daniel, A.; 33 S. First Avenue, Apartment 2, Highland Park, NJ 08904 (US).

(74) Agent: MACMASTERS, Thomas, L.; Fredrikson & Byron, P.A., 1100 International Center, 900 Second Avenue South, Minneapolis, MN 55402 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

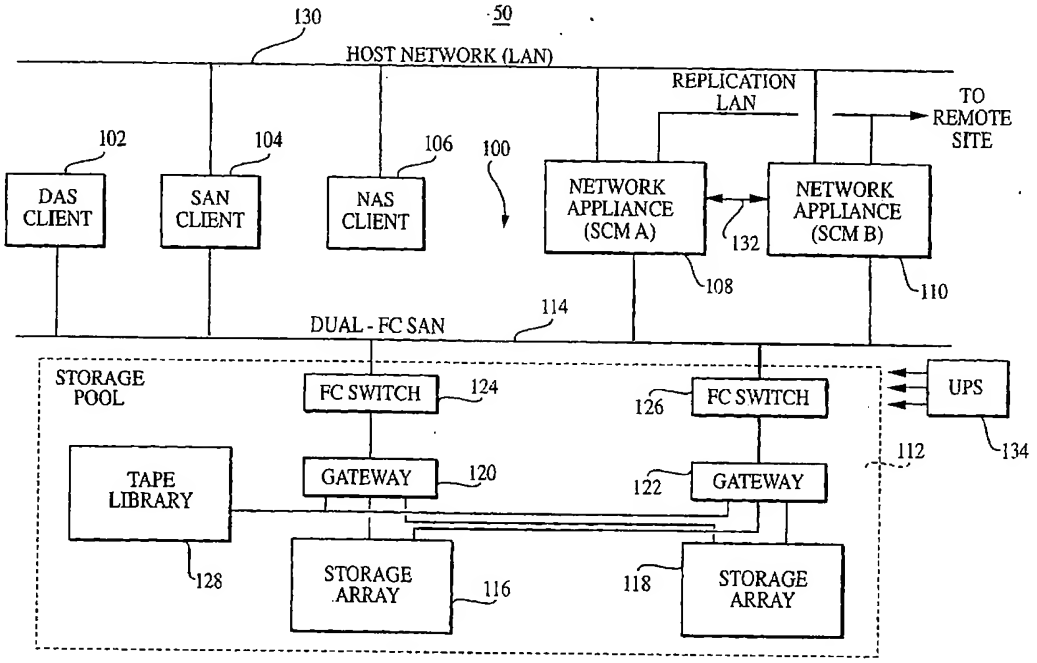
Published:
— *with international search report*

(88) Date of publication of the international search report:
28 November 2002

(15) Information about Correction:
Previous Correction:
see PCT Gazette No. 32/2002 of 8 August 2002, Section II

[Continued on next page]

(54) Title: NETWORK APPLIANCE



(57) Abstract: A method and apparatus for performing fault-tolerant network computing. The apparatus comprises a pair of network appliances coupled to a network. The appliances interact with one another to detect a failure in one appliance and instantly transition operations from the failed appliance to a functional appliance.

WO 01/082080 A3

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

INTERNATIONAL SEARCH REPORT

Int[illegible]nal Application No
PCT/US 01/12889

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06 G06F11/20

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 632 379 A (DIGITAL EQUIPMENT CORP) 4 January 1995 (1995-01-04) page 3, line 3 -page 5, line 20 page 7, column 8 -column 58	1-16
A	---	17
P,X	US 6 073 218 A (DEKONING RODNEY A ET AL) 6 June 2000 (2000-06-06) column 1, line 24 -column 5, line 65 column 7, line 36 -column 8, line 49	1-17
A	---	1,9,13
	US 5 944 838 A (JANTZ RAY M) 31 August 1999 (1999-08-31) column 1, line 12 -column 2, line 59 column 4, line 65 -column 7, line 64	

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 March 2002	18/03/2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Brichau, G

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

ırmation on patent family members

Int ›nal Application No
PCT/US 01/12889

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0632379 A	04-01-1995	EP 0632379 A2	04-01-1995
US 6073218 A	06-06-2000	AU 5701898 A WO 9828685 A1	17-07-1998 02-07-1998
US 5944838 A	31-08-1999	NONE	